제22회 한국정보처리학회 추계학술발표대회 논문집 제11권 제2호 (2004. 11)

유연한 프로세스 테일러링을 위한 프로세스 메타모델

김기호*, 홍선주*, 최성운*
*명지대학교 컴퓨터공학과
e-mail : parankiho@mju.ac.kr

A Process Metamodel for Flexible Process Tailoring

Ki-Ho Kim*, Sun-Joo Hong*, Sung-Woon Choi*
*Dept. of Computer Engineering, Myongji University

요 약

소프트웨어 개발 프로젝트의 특성을 고려한 프로세스의 테일러링은 고품질 소프트웨어 개발에 기본적인 요건이다. 테일러링을 유연하게 하기 위해서는 프로세스 구성 변화에 대한 영향이 지역화 되도록 프로세스 구성요소가 모듈화되어야 한다. 본 논문에서는 프로세스 모델을 모듈화된 요소기반으로 정의하기 위한 메타모델을 정의한다. 또한 정의된 메타모델을 기반으로 모듈화된 프로세스 모델의 예를 제시한다.

1. 서론

소프트웨어 개발 프로세스란 소프트웨어 개발 활동이 수행되는 순서나 방식이다.[1] 소프트웨어 시스템의 규모가 커지고 적용범위가 확대됨에 따라 소프트웨어 개발 프로세스도 다양한 형태로 발전되어왔다. 기존의 소프트웨어 개발 프로세스들은 다양한 시스템 개발에 활용되기 위해 일반적으로 필요한 모든 활동과 산출물을 포괄적으로 정의하고 있다. RUP(Rational Unified Process)[2], ISO/IEC12207[3]등은 포괄적인 활용이 가능한 프로세스의 대표적인 예이다. 이러한 프로세스들은 특정 프로젝트에 적합한 프로세스를 만들어 내기 위한 틀로써 이용이 가능하기 때문에, 프로세스 프레임워크[2], 표준 프로세스[3], 프로세스 모델[4]이라는 용어로 정의되기도 한다. 본 논문에서는 프로세스 모델이란 용어를 사용한다.

소프트웨어 개발 프로세스는 다양한 종류의 소프트웨어 개발 프로젝트들의 특성에 영향을 받는다.[5] 따라서 모든 소프트웨어 개발 프로세스는 기존의 프로세스 모델로부터 각 프로젝트의 특성에 맞게 테일러링되어야 한다. 테일러링이란 프로젝트의 여러가지 영향요소들을 고려하여 프로세스 모델의 일부를 삭제,수정하거나 새로운 요소를 추가하는 것과 관련된 모든 활동을 말한다. 이때 프로세스 모델 요소들간의 연관성은 유지되어야 한다.[6] 프로세스 테일러링 시에 유지되어야 하는 프로세스 모델 요소간의 연관성은 프로세스 메타모델로부터 정의된다. 프로세스 메타모델은 프로세스 모델을 구성하는 요소와 그들간의 관계, 규칙 등을 정의한다. OMG(Object Management Group)에서 표준화한 SPEM(Software Process Engineering Metamodel)[4]은 기존의 다양한 소프트웨어 개발 프로세스들을 바탕으로 정의된 프로세스 메타모델이다.

본 논문에서는 SPEM 에서 제시하는 프로세스 구성의 세가지 핵심요소인 역할, 산출물, 활동의 관계가 프로세스 테일러링의 유연성에 미치는 영향에 대해 분석한다. 또한 프로세스 테일러링의 유연성을 높이는데 가장 효과적인 프로세스 메타모델을 정의하고, 정의된 메타모델과 RUP 를 기반으로 프로세스를 모델링한다.

2. SPEM(Software Process Engineering Metamodel)

2.1 SPEM 모델링 4-계층 구조

SPEM 은 UML(Unified Modeling Language) 메타모델을 확장하여 소프트웨어 개발 라이프사이클 프로세스와 그 구성요소를 객체지향 적으로 정의하였다.[4] 그림 1 은 OMG 가 정의한 모델링의 4 계층 구조에서 SPEM 의 위치를 나타낸다. SPEM 은 메타모델 계층(M2)에 포함되며, RUP 와 같이 모델 계층(M1)에 속하는 소프트웨어 개발 프로세스 모델을 정의하기 위한

필수 구성요소 및 구조, 문법 등을 정의한다.

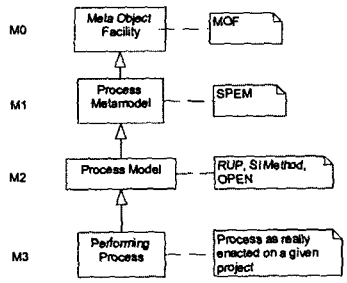

그림 1 OMG 의 모델링 4-계층구조

2.2 SPEM 의 개념모델

SPEM 은 역할(Role), 산출물(Work product), 활동(Activity)을 소프트웨어 개발 프로세스의 핵심요소로 정의한다. 산출물은 활동이 수행되기 위해 필요한 입력물이며 동시에 활동 수행 이후에 생산되는 출력물이다. 활동은 산출물 생산에 대한 책임을 갖는 추상적 개체인 역할에 의해 수행되는 행위이다.[6] SPEM 은 역할, 활동, 산출물 세가지 프로세스 구성요소를 서로 간의 연관성을 갖는 각각의 독립적인 개체로 정의한다.

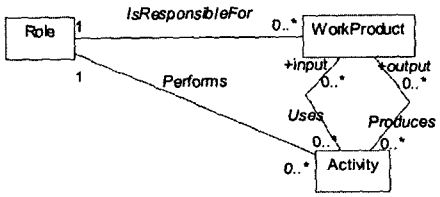

그림 2 SPEM 개념모델

역할은 산출물과 활동에 대해 각각 'IsResponsibleFor'와 'Performs'의 연관관계를 갖는데, 이 두 연관관계는 서로 상호제약적이다. 역할이 수행하는 활동의 종류에 따라 생산되는 산출물이 제한되며, 반대로 역할이 생산해야 할 책임이 있는 산출물에 따라 수행되어야 하는 활동이 제한된다. 그러나 SPEM 은 두 관계에 대한 상호제약성을 의미론적으로 명확하게 정의하고 있지 않다. 이처럼 밀접하게 연관된 두 관계의 상호제약성에 대한 내용 없이 산출물과 활동을 분리된 프로세스 구성요소로써 정의하는 것은 프로세스 테일러링의 유연성을 저하시키는 요인이 된다. 이 문제를 해결하기 위한 방법은 연관성이 높은 활동과 산출물을 다음 그림 3 과 같이 하나의 요소로써 모듈화하는 것이다.

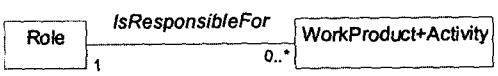

그림 3 산출물과 활동의 모듈화

모듈화는 산출물과 활동, 각 두 요소의 관점에서 이루어 질 수 있다. 다음 장에서는 두 가지 관점에서 프로세스를 모델링하기 위한 메타모델을 정의하고, 각 모델이 프로세스 테일러링의 유연성에 미치는 영향을 분석한다.

3. 관점별 프로세스 메타모델

3.1 활동 관점 프로세스 메타모델

활동 관점 프로세스 메타모델은 역할과 연관되는 프로세스 모델의 기본 구성요소가 활동을 중심으로 모듈화된다.

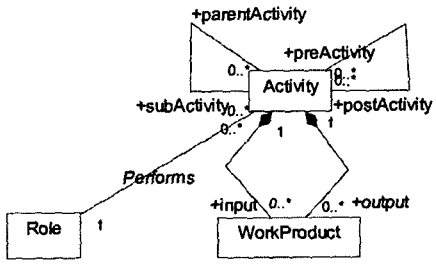

그림 4 활동 관점 프로세스 메타모델

산출물은 활동의 입력물과 출력물로써 활동 개체에 포함되며 활동간에는 선행활동과 후행활동의 관계 및 하위활동과 상위활동의 연관관계가 성립한다. 활동 관점 메타모델을 기반으로 프로세스 모델을 표현하면 다음과 같이 절차적인 형태로 나타난다.

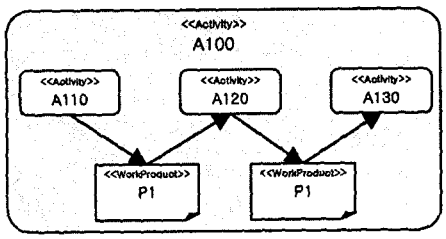

그림 5 활동 관점 프로세스 모델링

활동관점에서 절차적으로 정의된 프로세스의 구성체계는 프로세스 테일러링의 유연성 면에서 두가지 문제점을 갖는다. 첫번째 문제는 산출물의 입/출력으로 연결된 각 활동들의 강한 결합력이 프로세스 테일러링을 어렵게 한다는 것이다. 예를 들어 그림 6 과 같이 프로세스 테일러링을 위해 활동 A120 이 생략된다고 가정했을 때, 활동 A130 의 입력으로 사용될 산출물 P2 가 생성되지 않으므로 활동 A130 은 올바르게 수행되지 않을 것이다.

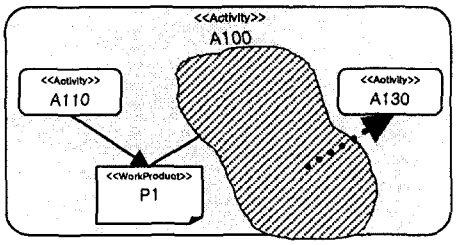

그림 6 활동관점 프로세스 모델의 테일러링

두번째 문제는 활동간 데이터 공유에서 발생한다. 일반적으로 소프트웨어 산출물은 내부에 많은 데이터들을 포함하고 있으며, 어떤 데이터들은 각기 다른 산출물에서 중복되어 나타난다. 이 경우 서로 다른 활동에 의해 접근되는 데이터의 일관성은 유지되기가 어렵다. 위 프로세스 모델은 산출물의 세부 항목의 구성이 가시적으로 나타나지 않으므로 활동과 연관된 산출물의 특정부분의 관리를 어렵게 한다.

기존의 절차적, 구조적 소프트웨어 개발 방법에서도 위와 동일한 문제가 발생하였으며, 소프트웨어 구성요소를 행위가 아닌 독립적인 객체 위주로 모듈화하여 해결하였다.[7] 마찬가지로 프로세스를 구성하는 기본요소를 독립적인 객체 위주로 모듈화 함으로써 해결책을 찾아 볼 수 있다.

3.2 산출물 관점 프로세스 메타모델

활동 관점의 프로세스 모델이 갖는 문제점을 해결하기 위해 독립적인 개체로써 식별성을 갖는 산출물을 프로세스 모델의 기본 구성요소인 클래스로 정의한다. 이때 산출물을 수정, 삭제, 생성하는 활동은 해당 산출물 클래스의 오퍼레이션으로 정의한다.

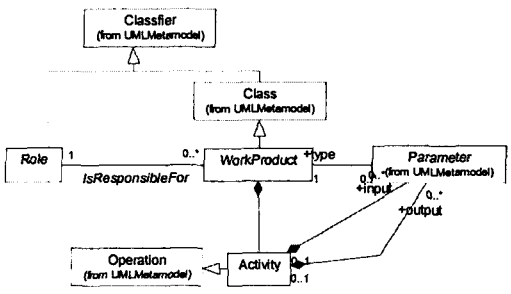

그림 7 산출물 관점 프로세스 메타모델

그림 7 은 산출물 관점 프로세스 메타모델의 개념을 나타낸다. WorkProduct 는 Activity 를 오퍼레이션으로 포함하는 클래스로 정의되며 Role 과 연관된다. 그림 7 의 개념모델을 상세화하면 그림 8 과 같다. 구체화된 메타모델에서는 WorkProduct 간에 공유하는 데이터의 일관성을 유지하기 위해 Item 을 기본 클래스로 정의하였다.

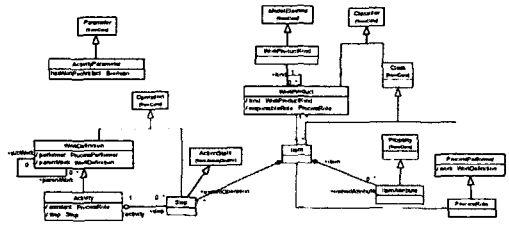

그림 8 구체화된 산출물 관점 프로세스 메타모델

산출물 중심의 프로세스 메타모델을 기반으로 정의된 프로세스 모델의 정적인 구조는 다음 그림과 같이 나타나며, 생명주기 프로세스는 산출물 객체간의 상호작용으로 표현된다.

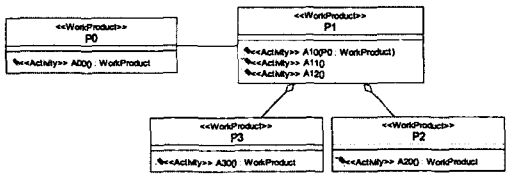

그림 9 산출물 중심 프로세스 모델의 정적구조

산출물 관점의 프로세스 모델링은 객체지향 모델링의 장점을 그대로 반영한다. 응집도가 높은 산출물 객체는 프로젝트의 특성에 따라 첨삭될 경우 다른 프로세스 구성요소에의 영향을 최소화 할 수 있어 테일러링이 유연해진다. [8]

산출물 작성과 활동 수행에 대한 책임을 갖는 역할은 그림 10 과 같이 관련 클래스와 연관된 물리적인 노드로 정의될 수 있다.

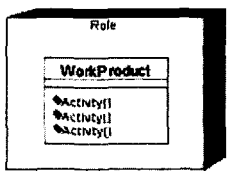

그림 10 역할과 산출물 연관

4. 메타모델 기반 객체지향 시스템 개발 프로세스 모델의 예

산출물 중심 프로세스 메타모델을 근간으로 하는 프로세스 모델의 개발은 해당 소프트웨어 개발 프로젝트에서 생성되어야 하는 산출물의 종류와 연관된 활동을 파악하는 것으로부터 시작된다. 본 논문에서는 객체지향 시스템 개발 프로세스 모델의 예를 UML 과 RUP 의 내용을 바탕으로 제시한다.

4.1 클래스 모델링

객체지향 시스템 개발 프로세스 모델을 구성하는 주요 클래스를 모델링하는 단계는 객체 및 클래스 정의와 클래스간의 관계, 그리고 클래스의 속성과 행위를 나타내는 단계이다. 이를 통해 프로세스 모델의 정적인 구조를 표현한다.

(1) 클래스 식별

객체지향 개발 산출물은 객체지향 시스템 개발 프로세스 모델의 클래스를 정의하기 위해 필요한 기본 요소로써 UML 모델을 기반으로 프로세스 모델 클래스를 정의한다. UML 에서 정의한 모델 이외에도 유스케이스 분석을 위해 요구되는 요구사항명세서와 각 UML 모델들에 대한 명세서도 프로세스 모델 클래스 후보가 된다.

(2) 클래스 관계 및 속성 정의

위에서 설명한 객체지향 모델들을 기반으로 클래스를 개발하기 위해서는 각 모델들을 구성하고 있는 요소와 그들 간의 관계를 분석해야 한다. UML 메타모델은 각 모델들의 시멘틱을 정의한다. UML 메타 모델은 UML 다이어그램의 의미를 나타내기 위해서, 각 다이어그램의 구성요소를 클래스로 나타내고, 각 구성요소간의 관계를 클래스간의 관계를 표현함으로 나타내고 있다.

(3) 클래스 오퍼레이션 정의

프로세스 모델 클래스의 오퍼레이션을 정의하기 위해서 RUP 의 활동들을 고려한다. RUP 는 개발활동을 프로세스 컴포넌트, 활동, 작업 등으로 계층화 하는데, 활동은 여러 개발모델의 생성에 영향을 미치기 때문에, 활동을 기반으로 프로세스 모델 클래스의 오퍼레이션을 정의하는 것은 적합하지 못하다. 본문에서는 요구사항 획득, 분석&설계 단계 중 필수적인 세부 작업단위를 중심으로 프로세스 모델 클래스의 오퍼레이션을 정의한다. 클래스의 오퍼레이션은 산출물 클래스의 속성을 사용, 변화시키는 행위를 중심으로 정의된다.

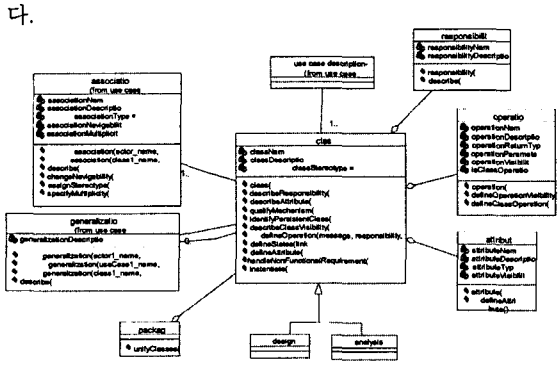

그림 11 클래스 모델의 예

4.2 프로세스 생명주기 모델링

소프트웨어 개발 생명주기는 프로세스 모델을 구성하는 산출물 객체들간의 상호작용을 모델링 함으로써 정의된다. 상호작용 모델링의 단계는 먼저 RUP 의 활동을 사용하여 각 프로세스에 대한 시나리오를 작성하고, 작성된 시나리오에 따라 프로세스 모델 객체간의 상호작용 다이어그램을 작성하였다.

4.3 배치 모델링

3 장에서 설명한 SPEM 확장모델을 기반으로 RUP 에서 정의된 작업자의 책임(responsibility)에 적합한 프로세스 모델 객체를 할당하여 실제로 업무 분담을 정의한다. 이러한 물리적 배치를 기술하는 기법으로 UML 에서 제시하는 배치 다이어그램을 사용하면, 시스템 개발 조직에 속한 작업자들을 노드(node)로 각 방법론 객체는 해당 노드에 할당되는 프로세스(process)로 매핑 된다. RUP 의 각 작업자의 책임을 분석하여 유사성이 높은 작업자끼리 묶어 상위 구조를 형성하여 작업자를 추상적으로 계층화함으로써 해당 조직의 실제 인원수에 따라 조직 내 업무 분담을 유연하게 할 수 있도록 하였다.

5. 결론

OMG 에서 표준화한 소프트웨어 개발 프로세스 메타모델인 SPEM 을 기반으로 하는 산출물 중심 프로세스 메타모델은 프로세스 구성요소들을 독립적인 객체로서 정의하여, 이를 클래스화, 계층화함으로써 일반 개발 프로세스에서 유추하여 재사용될 수 있다. 메타모델이 제공하는 문법과 의미론은 개발 프로세스를 지원하는 도구에 표준화된 인터페이스와 저장 메커니즘 등의 핵심요소를 제공하여 프로세스 적용 경험의 축적 및 유지보수를 용이하게 할 것이다. 특히 시스템 개발 시에 필요한 산출물을 중심으로 개발 프로세스의 저작(Authoring) 및 테일러링 툴 개발을 위한 기반으로써 활용될 수 있다.[4] 프로세스 메타 모델의 추후 연구는, 프로세스 모델링과 테일러링을 실제 개발에 적용하여 나타나는 결과의 분석과 그에 따른 보완을 포함 한다. 특히 도메인 분석을 통한 프로세스 테일러링의 체계화도 진행 되어야 한다.

참고문헌

[1] Roger S. Pressman. *Software Engineering, A Practitioner's Approach, 5th Ed.* McGraw-Hill, 2001.

[2] Philippe Kruchen. *The Rational Unified Process.* Addison Wesley Longman Inc.,1999.

[3] Singh, R., *An Introduction to International Standard ISO/IEC12207.* 1999.

[4] OMG , Software Process Engineering Metamodel Specification 1.0. 2002.

[5] Watts S. Humphrey. *Managing the Software Process.* Addison-Wesley, 1989.

[6]Il-Chul Yoon, Sang-Yoon Min, and Doo-Hwan Bae. "Tailoring and Verifying Software Process". In *Proceedings of APSEC'01*, 2001.

[7] Timothy Budd. *An Introduction to Object-Oriented Programming.* Addison-Wesley, 1991.

[8] M.I.Kellner. "Connecting reusable software process elements and components". In *Proc. International Software Process Workshop*, 1996.