

# 사용 관점 중심의 컴포넌트 모델링을 위한 UML의 확장

김경민<sup>○</sup> 김태웅 김태공  
인제대학교 전산학과  
{kmkim<sup>○</sup>, twkim, ktg}@cs.inje.ac.kr

## Extension of UML for Components Modeling Based on Viewpoint of Component Use

Kyung-Min Kim<sup>○</sup> Tae-Woong Kim Tae-Gong Kim  
Dept. of Computer Science, Inje University

### 요약

컴포넌트 모델링 방법은 컴포넌트 개발에서 시스템을 이해하고 분석하기 위해 중요한 부분을 차지하고 또한 재사용성을 높이는 방안으로써 받아들여지고 있다. 컴포넌트는 일반적으로 컴포넌트를 개발하는 사람과 그것을 조립하여 사용하는 사람이 다르며 이에 따른 내용과 목적이 다르기 때문에 대규모의 컴포넌트 시스템을 개발하는데 있어서 그 역할에 따라 두 가지 유형으로 구분하는 것이 중요하다. 이것이 현실화되기 위해서는 명확한 컴포넌트와 인터페이스 명세가 필요하며 조립자와 개발자 간의 서로 다른 관점에서의 컴포넌트 모델이 필요하다. 이에 본 논문에서는 조립자 관점과 생성자 관점이라는 서로 다른 역할에 따라 두 가지 유형의 컴포넌트 모델을 제안하며 이를 위해 UML을 확장한다. 그리고 이를 구매패주 시스템에 적용하여 그 효용성을 검토한다.

### 1. 서론

소프트웨어 컴포넌트란 하나 이상의 기능을 갖는 독립적인 소프트웨어이며, 조립을 통해 응용프로그램을 작성할 수 있는 부품 형태의 소프트웨어를 말한다[1]. 이런 컴포넌트 개발을 위해서는 시스템을 이해하고 분석하기 위한 컴포넌트 모델링 방법이 중요한 부분을 차지하고 재사용성을 높이는 방안으로써 받아들여지고 있다. 여기에서 모델링은 대상을 종합적으로 이해하고 주어진 조건이나 입력에 대한 대상의 반응을 예측할 목적으로 비교적 엄격한 규칙에 따라 구현된 표현을 의미한다[2]. 컴포넌트 모델링 방법은 개발 초기 단계의 부정확한 산출과 불충분한 모델 구축의 문제를 해결할 수 있는 방법으로 간주되고 있으며, 개발 노력과 유지보수 비용을 줄일 수 있다.

이러한 컴포넌트 기반의 모델에는 UML[3]을 기반으로 하는 RUP[4], Catalysis[5], Advisor[6], MaRMI-III[7] 등 소프트웨어 개발 방법론에서 제공하는 모델들이 있다. 컴포넌트는 일반적으로 컴포넌트를 개발하는 사람과 그것을 조립하여 사용하는 사람이 다르며 이에 따른 내용과 목적이 다르기 때문에 대규모의 컴포넌트 시스템을 개발하는데 있어서 그 역할에 따라 두 가지 유형으로 구분하는 것이 중요하다. 하지만 기존의 개발 방법론에서는 이런 특성을 적절하게 반영하기 어려우며 또한 컴포넌트 기반 개발 프로세스의 각 작업에 대하여 체계적이면서 구체적인 지침들을 충분히 제시하지 않고 있다[8].

이에 본 논문에서는 효율적인 컴포넌트 기반의 개발을 위해 조립자 관점과 생성자 관점이라는 서로 다른 역할에 따라 두 가지 유형의 컴포넌트 모델을 제안한다. 조립자 관점의 컴포넌트 모델에서는 컴포넌트를 조립할 수 있는 컴포넌트 인터페이스 정보를 표현하고 생성자 관점의 컴포넌트 모델에서는 실제 구현에 필요한 컴포넌트 전체 상호작용 관계와 서비스의 내부 비즈니스 로직에 관한 정보를 표현한다. 이를 위해 UML을 확장하고 구매패주 시스템에 적용하여 그 효용성을 검토한다.

### 2. 기존 컴포넌트 모델들의 문제점 및 해결 방안

#### 2.1 기존 컴포넌트 모델들의 문제점 분석

RUP, Catalysis, Advisor, MaRMI-III 방법론에서 제공하는 컴포넌트 관련 모델들은 UML을 기반으로 모델들을 표현하고 있으며 개발 과정의 여러 단계들에서 컴포넌트 개념을 나타내

고 있다. 때문에 객체 지향 중심인 UML의 일반적인 단점을 가지게 되어, 내부 디자인의 규모가 객체지향 방법론의 중심이 되는 객체 혹은 클래스와 컴포넌트 사이에 큰 차이점을 보이게 되므로 정확한 컴포넌트의 설계 및 표현에 한계가 있다. 또한 컴포넌트의 기본적인 특성인 인터페이스에 대한 표현이 단순하고 제한된 표현 방법으로 그 뜻을 나타내기 때문에 여러 가지로 해석이 가능할 수 있는 모호한 점이 있다.

이들 방법론에서는 개발 방법의 단계마다 여러 모델들로 분산되어 컴포넌트 개념을 기술하고 있기 때문에 컴포넌트 구현을 위한 내부 정보를 얻기 위해서는 컴포넌트 모델에서 역으로 방법론 각 단계들에서 제시하는 모델들을 추적하여 찾아야 한다. 이것은 컴포넌트 모델이 방법론의 한 단계로써 포함하고 있는 정보들이 추상적이기 때문이다. 이 같은 모델 추적은 방법론에 대한 기반 지식이 없는 개발자에게는 힘든 작업이며, 이것은 다양한 컴포넌트 모델 정보를 재사용하기 위해서는 개발자가 모든 방법론을 숙지해야 한다는 것을 의미한다.

컴포넌트 개발은 크게 컴포넌트 조립에 의한 개발과 컴포넌트 생성에 의한 개발로 이루어지는데, 기존 모델에서는 개발되는 방법에 따른 조립과 생성을 위한 컴포넌트 모델의 구분 없이 표현한다. 때문에 컴포넌트 모델 정보의 전부가 컴포넌트를 조립하는 사람과 관련을 갖는 것은 아니며, 실제 이 모델을 기반으로 구현하려는 사람에게서는 부족한 정보들이 많이 된다.

#### 2.2 문제점 해결 방안

본 논문에서는 컴포넌트가 사용되는 역할에 따라 컴포넌트를 조립하여 개발하는 조립자 관점과 직접 컴포넌트를 구현하여 개발하는 생성자 관점의 두 가지 유형으로 컴포넌트 모델을 제안한다. 그리고 모델 추적이라는 힘든 작업을 해결하고자 방법론 과정 중 한 단계에서의 모델이 아닌 독립된 컴포넌트 모델을 제시하여, 제안하는 모델들이 조립에 의한 컴포넌트 개발과 생성에 의한 컴포넌트 개발에 실질적으로 사용될 수 있도록 한다.

이를 위해 컴포넌트 조립을 위한 정보와 컴포넌트 생성을 위한 정보로써 표 1과 같이 정의한다. 조립을 위한 정보에는 해당 컴포넌트에서 제공하는 서비스를 직접 호출하여 사용할 수 있는 컴포넌트 인터페이스 정보를 나타낸다. 생성을 위한 정보

에는 실제 컴포넌트를 구현하기 위한 컴포넌트의 전체적인 구조와 컴포넌트들 간의 상호작용 정보 그리고 해당 컴포넌트의 하나의 인터페이스에 대한 내부 상세 정보로서 오퍼레이션들과 속성들, 그들 타입간의 관계 정보를 나타낸다.

이들 정보에 따른 조립자 관점의 컴포넌트 모델과 생성자 관점의 컴포넌트 모델은 3.1과 3.2에서 제안한다.

표 1. 컴포넌트 조립과 생성을 위한 정보

조립을 위한 정보	컴포넌트 명, 인터페이스 명, 인터페이스의 오퍼레이션 명, 오퍼레이션 인자타입, 오퍼레이션 리턴타입
생성을 위한 정보	컴포넌트 명, 컴포넌트 간의 관계, 인터페이스 명, 인터페이스의 오퍼레이션 명, 오퍼레이션 인자타입, 오퍼레이션 리턴타입, 인터페이스의 속성 타입, 인터페이스의 외부 인터페이스와의 관계, 인터페이스의 내부 타입(객체) 명, 타입 간의 관계, 타입의 오퍼레이션 명, 타입의 오퍼레이션 인자타입, 타입의 오퍼레이션 리턴타입, 타입의 속성 타입

3. UML의 확장

3.1 조립자 관점의 컴포넌트 모델

컴포넌트간의 조립은 인터페이스를 기반으로 결합되며 이 인터페이스는 정보는닉 원리에 따라 컴포넌트를 자유롭게 대체하고 독립적으로 사용할 수 있도록 지원한다.

조립자 관점의 컴포넌트 모델에서는 컴포넌트에서 제공하는 서비스를 직접 호출하여 사용할 수 있도록 컴포넌트 인터페이스 정보를 표현하기 위해 표 2와 같이 UML을 확장하며 정의된 모델은 그림 1과 같다.

이 모델은 해당 컴포넌트가 어떤 컴포넌트와 관련 되어 수행되는지 알 필요 없이 단순히 컴포넌트를 호출해서 사용가능하도록 조립자가 알아야 하는 컴포넌트 명과 인터페이스 정보만을 나타낸다. 이것은 자세한 비즈니스 로직은 숨긴 채 컴포넌트 인터페이스를 통한 서비스를 가능하게 한다.

표 2. 조립자 관점의 컴포넌트 모델을 위한 UML 확장

UML 확장	포함 정보	의미
	컴포넌트 명	사용할 컴포넌트를 의미함
	인터페이스 명, 오퍼레이션 명/인자타입/리턴타입	컴포넌트 사용 시 필요한 인터페이스의 오퍼레이션 목록으로 해당 컴포넌트에서 제공하는 서비스를 의미함

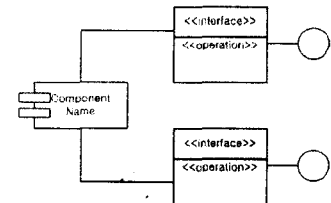


그림 1. 조립자 관점의 컴포넌트 모델

3.2 생성자 관점의 컴포넌트 모델

생성자 관점의 컴포넌트 모델은 순수 구현을 위한 개발자에게 필요한 많은 정보들을 나타낸다. 컴포넌트의 인터페이스 구현물이 자신의 책임을 완수하기 위해 다른 컴포넌트와 어떤 상호작용을 하는지, 컴포넌트에서 제공하는 서비스의 내부 비즈니스 로직이 어떠한지 내부 로직들 간의 관계가 어떠한지, 내부적으로 사용되는 정보들은 어떠한지에 관해서 표현한다.

이처럼 실제 생성자에게 필요한 모든 정보를 포함하기 위해 컴포넌트 상호작용 모델과 인터페이스 내부 모델 두 단계로 구분하며, 이 두 단계의 모델에 구체적으로 나타냄으로써 실제 구현 시 모델의 이용 효율을 높인다.

■ (1단계) 컴포넌트 상호작용 모델

컴포넌트 상호작용 모델은 생성자를 위한 1단계 모델로서 개발하려는 컴포넌트의 전체적인 구조와 컴포넌트들 간의 상호작용 정보를 나타낸다. 이를 위해 표 3과 같이 UML을 확장하며 정의된 모델은 그림 2와 같다.

표 3. 생성자의 컴포넌트 상호작용 모델을 위한 UML 확장

UML 확장	포함 정보	의미
	컴포넌트 명	사용할 컴포넌트를 의미함
	인터페이스 명, 오퍼레이션 명/인자타입/리턴타입	컴포넌트 사용 시 필요한 인터페이스의 오퍼레이션 목록으로 해당 컴포넌트에서 제공하는 서비스를 의미함
	.	화살의 출발지 컴포넌트에서 목적지의 컴포넌트를 필요로 함을 의미

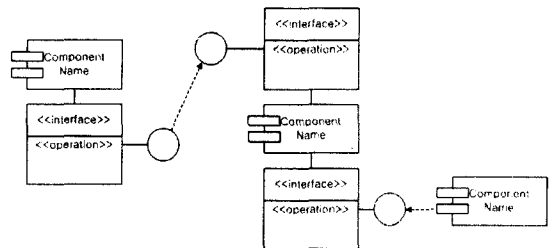


그림 2. 생성자의 컴포넌트 상호작용 모델

■ (2단계) 인터페이스 내부 모델

생성자를 위한 2단계 모델인 인터페이스 내부 모델은 해당 컴포넌트의 하나의 인터페이스에 대한 내부 상세 모델로서 실제 컴포넌트의 인터페이스를 구현하기 위한 오퍼레이션들과 속성들, 그들 타입간의 관계 정보를 나타낸다. 이를 위해 표 4와 같이 UML을 확장하며 정의된 모델은 그림 3과 같다.

표 4. 생성자의 인터페이스 내부 모델을 위한 UML 확장

UML 확장	포함 정보	의미
	컴포넌트 명, 인터페이스 명, 속성명/인자타입/리턴타입, 내부 타입 관계도	해당 컴포넌트의 하나의 인터페이스에 대한 상세 정보 나타내는 기본 틀로서 생성자에게 필요한 인터페이스 내부 정보와 로직 간의 타입 관계도를 포함함.
	타입 명, 타입에 대한속성정보, 오퍼레이션명/인자타입/리턴타입	특정 인터페이스 내에서 내부적으로 처리되는 로직에 관한 정보로서 추상화 높은 타입을 의미함
	컴포넌트 명, 인터페이스 명, 반복 표시자(1, *)	관련 있는 외부 인터페이스와 관계를 지정함. 동일 컴포넌트 내의 다른 인터페이스이거나 다른 컴포넌트의 인터페이스일 수 있음
	반복 표시자(1,*)	타입 관계도에서 타입간의 연관 관계가 있음을 의미

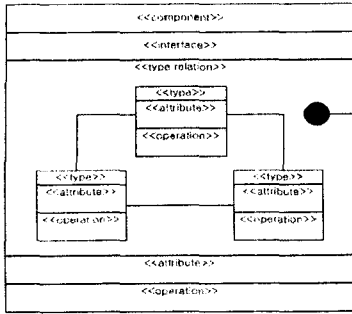


그림 3. 생성자의 인터페이스 내부 모델

4. 적용 사례

본 연구에서는 사용 관점에 따른 컴포넌트 모델을 생성하고 저장, 관리하며 이들 정보를 다시 컴포넌트 사용자에게 새로운 응용 개발에 필요한 요소로서 재사용될 수 있도록 컴포넌트 모델 저장소를 개발하고 있다. 현재 확장된 UML을 이용하여 사용 관점에 따른 컴포넌트 모델을 생성하고 생성된 모델들을 XML기반으로 저장할 수 있으며, 컴포넌트 모델 분류와 모델 검색 부분들에 관한 연구와 개발들이 진행 중에 있다.

이 컴포넌트 모델 저장소에서 조립자와 생성자 관점의 컴포넌트 모델을 구매발주 시스템 중 구매 컴포넌트로 실제 적용해 보았다.

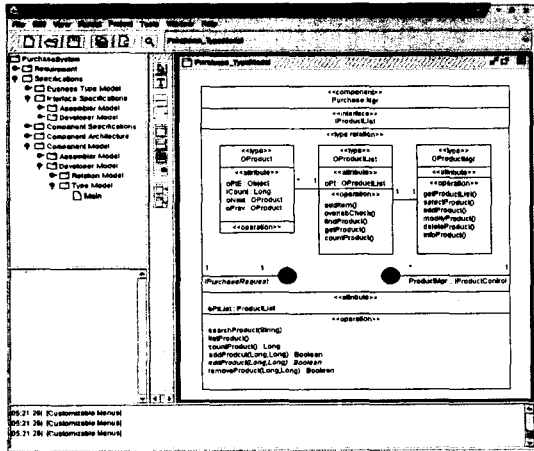


그림 4. PurchaseMgr의 생성자의 인터페이스 내부 모델

그림 4는 구매 컴포넌트(PurchaseMgr)의 상품리스트 인터페이스(IProductList)에 대한 2단계 인터페이스 내부 모델을 나타낸다. 실제 구현에 필요한 구체적인 정보들으로써 외부에서 사용이 가능한 오퍼레이션(operation)들과 속성(attribute) 정보들, 내부 로직을 처리하는 타입(type)정보들과 그들 간의 타입 관계(type relation)를 포함한다. 그리고 내부 인터페이스와 외부 인터페이스와의 관계도 명시한다.

5. 결론

본 연구에서는 RUP, Catalysis, Advisor, MaRMI-III 개발 방법론에서 나타내는 기존 컴포넌트 관련 모델들의 문제점을 분석하여, 이에 대한 해결 방안으로써 사용 관점 중심의 컴포넌트 모델링을 위해 UML을 확장하였다. 그리고 본 연구에서 개발 중인 컴포넌트 모델 저장소에서 그 사례를 적용해 보았다.

기존 방법론들에서는 개발 방법 각 단계마다 여러 모델들로 분산되어 컴포넌트 개념을 기술하고 있기 때문에 컴포넌트 내부 정보를 얻기 위해서는 방법론 각 단계들에서 제시하는 모델들을 역 추적하여 찾아야 했다. 이것은 컴포넌트 모델이 방법론의 한 단계로써 포함하고 있는 정보들이 추상적이기 때문이며, 이 같은 모델 역 추적은 방법론에 대한 기반 지식이 없는 개발자에게는 힘든 작업이다. 이것은 다양한 컴포넌트 모델 정보를 재사용하기 위해서는 개발자가 모든 방법론을 숙지해야 한다는 것을 의미한다.

그리고 컴포넌트 개발은 크게 컴포넌트 조립에 의한 개발과 컴포넌트 생성에 의한 개발로 이루어지는데 기존 모델에서는 개발되는 방법에 따른 이 같은 모델의 구분 없이 표현하기 때문에, 그 컴포넌트 모델 정보의 전부가 컴포넌트를 조립하는 사람과 관련을 갖는 것은 아니며 또한 실제 이 모델을 기반으로 구현하려는 사람에게는 부족한 정보들이 많게 되었다.

이에 본 논문에서 컴포넌트가 사용되는 역할에 따라 컴포넌트를 조립하여 개발하는 조립자 관점과 직접 컴포넌트를 구현하여 개발하는 생성자 관점의 두 가지 분리된 유형으로 컴포넌트 모델을 제안했다. 그리고 모델 추적이라는 힘든 작업을 해결하고자 방법론에서 제공하는 하나의 단계에서의 모델이 아닌 독립된 컴포넌트 모델을 제안하여, 이 모델만을 이용하여 조립에 의한 컴포넌트 개발과 생성에 의한 컴포넌트 개발에 실질적으로 사용될 수 있도록 하였다.

이를 통해 조립자는 컴포넌트 내부의 자세한 비즈니스 로직은 숨긴 채 컴포넌트 인터페이스를 통한 서비스가 가능하게 되었으며, 생성자는 필요한 모든 정보를 두 단계의 모델로 구체적으로 나타냄으로써 실제 구현 시 모델의 이용 효율을 높였다. 또한 이렇게 분리된 두 가지 유형으로 정의함으로써 생성자 계약조건에 대한 변경이 조립자 계약의 변경을 수반하지 않게 되므로 조립자에게 어떠한 영향도 없이 컴포넌트 변경에 쉽게 대응할 수 있도록 하였다.

본 논문에서 제안하는 컴포넌트 모델에서는 모델이 사용되는 관점에 따라 필요한 컴포넌트의 정보들을 적합하게 구분 하였으나 컴포넌트의 기능적인 정보들만을 표현하고 있다. 앞으로 컴포넌트들 간의 제어 정보와 컴포넌트 오퍼레이션들의 내부 알고리즘 정보들의 표현에 대한 연구가 진행되어야 하겠다.

참고문헌

- [1] Dedmond F. D'Souza, Alan C. Wills, "Object, Component and Frameworks With UML", ADDISON-WESLEY, 1998
- [2] 김창완, 객체지향 모델링과 구현, 1998
- [3] Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide", Addison Wesley, 1999
- [4] Kruchten, P. B., "Rational Unified Process, The: An Introduction , Second Edition", Addison-Wesley, 2000 <http://www.rational.com>
- [5] D'Souza, D. F., and A. C. Wills, "Object, Components, and Frameworks with UML: The Catalysis Approach", Addison-Wesley, 1998 <http://www.catalysis.org>
- [6] Dodd, J., et al., "Advisor 2.04", Sterling Software, 1999 <http://www.cai.com>
- [7] "마르미 방법론", 한국 전자 통신 연구원, at URL: <http://www.component.or.kr>
- [8] 김수동, "컴포넌트 정의 및 관련 기술 동향", 소프트웨어 공학회지 12권 3호 pp.5-18, 1999