
UML 클래스 도해의 처리도구를 위한 메타데이터의 정보 구축

김재훈* · 김윤호*

*안동대학교

Information of Metadata Structure for an Authorization Tool of UML Class Diagram

Jae-Hoon Kim* · Yun-Ho Kim*

*Andong National University

E-mail : zoom50210@gmail.com, unokim@andong.ac.kr

요 약

본 논문에서는 UML 클래스 도해의 저작도구를 위한 메타데이터의 정보 구축을 제시하고자 한다. UML의 클래스 다이어그램에서 클래스를 모델링 할 때, 표현되는 클래스(Class)와 관계(Relationship)를 정의 하였다. 클래스는 어떠한 사물의 개념적인 것을 나타내며, name, attribute, operation 세 가지로 정의하였다. 관계는 클래스와 클래스 사이의 관계를 뜻하며, 관계의 이름, From클래스, To클래스, 관계의 유형을 정의한다. 그리고 UML 클래스 도해의 처리도구를 위한 메타데이터의 정보를 가공하고 처리하는 방법을 제안하였다.

ABSTRACT

This paper presents the establishment of the information of metadata for an Authorization tool of UML class diagram. When it comes to modeling classes in UML class diagram, the paper defines expressed classes and the relationship. The class represents the concept of an object, which defines Name, Attribute and Operation. The relationship is between classes, which defines the name and the type of the Relationship, From Class and To Class. And it suggests how to handle and process the information of metadata for an authorization tool of UML class diagram.

키워드

UML, Class Diagram, Class, Relationship

1. 서 론

OMG(Object Management Group)에서[1] 1997년 UML을 발표했다. UML의[2][3] 목표 중 하나가 개발 커뮤니티에 안정적이고, 일반적인 그래픽 언어를 제공하는 것이다. UML은 수년 동안 바라던 통합된 표준 모델링 표기법을 탄생시켰다. UML을 이용하면 시스템 구조와 디자인 계획을 읽을 수 있고 분산 시킬 수 있다. 설계의 청사진을 보는 것처럼 사용된다. 그리고 UML은 방법론이 아니기 때문에 어떠한 형식적인 작업 생성물들이 필요 없다. 하지만 시스템을 설계할 때 여러 가지 다이어그램을 제공한다. 고객과 개발자 또는 개발자와 개발자간에 의사소통을 원활 하게 할

수 있도록 해준다.

UML에서 제공하는 많은 다이어그램 중에서 class diagram은 시스템의 구조를 표현하는데 가장 적합하며 큰 영향을 준다. class diagram에서 클래스(Class)와 관계(Relationship)를 나눌 수 있다. 클래스는 name, attribute, operation을 나타낼 수 있으면 관계는 UML에서 정의한 의존(dependency), 연관(association), 일반화(generalization), 실체화(realization), 집합(aggregation), 합성(composition)을 나타낼 수 있다. 클래스는 객체를 정의하는 것이며, 관계는 클래스와 클래스 사이에 관계를 나타내기 위한 것이다.

II. UML 클래스 도해의 메타데이터 정보 구축

UML의 클래스 다이어그램은 객체지향 시스템을 모델링 할 때, 가장 일반적으로 사용하게 된다. 시스템의 정적인 설계 뷰를 모델 할 때, 클래스 다이어그램을 사용한다. 대부분 시스템의 용어를 가지고 모델링을 하거나, 협력 모델링, 또는 스키마 모델링하는 것을 포함한다. 클래스 다이어그램의 가장 큰 특징은 클래스의 집합, 클래스들의 관계를 표현하는 것이다. 아래의 그림 1과 같이 클래스다이어그램의 구성을 나타내고 있다.

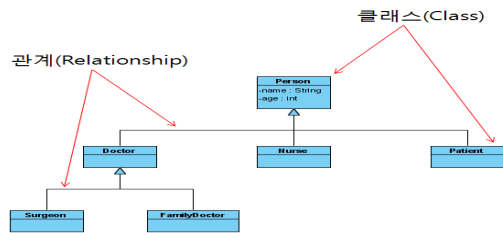


그림 1. 클래스 다이어그램의 구성도

클래스 다이어그램은 클래스(Class)와 관계(Relationship)를 표현할 수 있다. 클래스(Class)는 시스템의 용어를 획득하여 클래스를 표현 할 수 있으며, 소프트웨어 그리고 하드웨어, 또는 어떠한 사물의 개념적인 것을 표현한 것이다. 관계(Relationship)를 표현 하는 것은 시스템에서 어떠한 기능을 수행하기 위해서 클래스들끼리 협력하여 수행하는 경우가 있는데, 이러한 경우 클래스들 간에 협력하기 위해서 클래스와 클래스 사이에 관계를 나타낼 수 있다.

2.1 클래스의 메타데이터 정보 구축

UML의 클래스다이어그램에서 클래스를 모델링 할 때, 클래스는 이름(name)과 속성(attributes), 행위(operations)를 정의하고 있다. 이름은 클래스의 이름을 뜻하며, 속성(attribute)은 클래스의 객체 상태를 표현하거나 상태를 표현하기 위한 데이터의 종류를 말하며 한 개 이상의 속성을 가지거나 속성이 없을 수 있다. 행위(operation)는 클래스의 행위(behavior)를 뜻하며 속성과 마찬가지로 한 개 이상의 행위를 가지거나 없을 수 있다. 그림 2는 사각형의 첫 번째 칸에 클래스의 name, 중간 칸에 attribute, 세 번째 칸에 operation을 표현하고 있다.

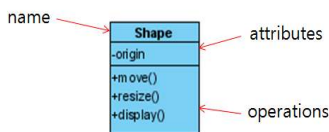


그림 2. 클래스의 구조

2.1.1 Class

클래스의 메타데이터 정보 정의는 클래스의 가시성(classVisibility), 이름(name), 속성(attribute), 행위(operation) 이렇게 네 가지로 구분하여 나타내었으며, 표1과 같다. 클래스의 이름은 각 클래스마다 하나만 존재하고 유일하기 때문에 각각의 클래스의 정보를 찾기 위해서 클래스이름으로 식별할 때 사용한다. 그러므로 클래스 이름은 다른 클래스와 이름이 중복되면 안 된다.

표 1. 클래스의 메타 정보 정의

이름	설명
classVisibility	클래스의 가시성 표현
name	클래스의 이름 표현
attribute	클래스의 속성 표현
operation	클래스의 행위 표현

2.1.2 Attribute

클래스는 속성(attribute)을 하나 이상 가지거나 속성이 없을 수 있기 때문에 하나 이상의 속성을 가지는 경우를 위해서 배열에 속성을 가지고 있도록 한다. 속성의 메타데이터 정보 정의는 가시성(attributeVisibility), 이름(attributeName), 타입(attributeType) 이렇게 세 가지로 구분하여 나타내고 표2와 같다.

표 2. 속성(attribute)의 메타데이터 정보 정의

이름	설명
attributeVisibility	속성의 가시성 표현
attributeName	속성의 이름 표현
attributeType	속성의 타입 표현

2.1.3 Operation

행위(operation) 또한 클래스가 하나 이상 가지거나 없을 수 있기 때문에 하나 이상의 속성을 가지는 경우를 위해서 마찬가지로 배열에 행위를 가지고 있다. 행위의 메타데이터 정보 정의는 가시성(operationVisibility), 이름(operationName), 리턴 타입(operationReturnType), 파라미터(operationParameter) 이렇게 네 가지로 표현했으며 표 3과 같다.

표 3. 행위(operation)의 메타데이터 정보 정의

이름	설명
operationVisibility	행위의 가시성 표현
operationName	행위의 이름 표현
operationReturnType	행위의 리턴타입
parameter	행위의 파라미터 표현

2.1.3.1 Parameter

마찬가지로 하나의 행위에 파라미터(parameter)가 하나 이상 가지거나 안 가질 수 있기 때문에 배열에 넣어서 표현한다. 파라미터의 메타데이터 정보 정의는 파라미터 이름(parameterName), 파라미터 리턴타입(parameterReturnType) 두 가지로 나타내었고 표4와 같다.

표 4. 파라미터의 메타데이터의 정보 정의

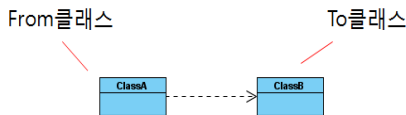
이름	설명
parameterName	파라미터의 이름 표현
parameterReturnType	파라미터의 리턴타입 표현

2.2 관계의 메타데이터 정보 구축

UML에서 관계(Relationship)는 6가지의 관계를 정의하고 있다. 의존(dependency), 연관(association), 일반화(generalization), 실체화(realization), 집합(aggregation), 합성(composition) 이렇게 여섯 가지의 관계를 나타머 표5와 같다.

표 5. UML에서 관계(Relationship) 정의

관계	표기법
의존(dependency)	꺼져 화살표에 점선
연관(association)	실선
일반화(generalization)	속이 검은 삼각형에 실선
실체화(realization)	속이 빈 삼각형에 점선
집합(aggregation)	속이 빈 다이아몬드에 실선
합성(composition)	속이 검은 다이아몬드에 실선



관계(Relationship)의 유형
그림 3. 관계의 구조

관계(Relationship)의 메타데이터 정보 정의는 관계의 이름(relationshipName), From클래스(fromClass), To클래스(toClass), 관계의 유형(relationshipType) 이렇게 네 가지로 정의 하였다. 관계의 이름은 각각의 관계를 식별하기 위해서 정의 하였고, From클래스는 관계의 시작 클래스를 뜻하며, To클래스는 관계의 도착 클래스를 뜻한다. 그리고 마지막으로 클래스와 클래스 사이의 관계를 정의하는 관계 유형을 정의하고 표 6과 같다.

표 6. 관계의 메타데이터 정보 정의

이름	설명
relationshipName	관계의 이름 표현
fromClass	시작 클래스 표현
toClass	도착 클래스 표현
relationshipType	관계의 유형 표현

III. UML 클래스 도해의 메타데이터 정보 처리

이 장에서는 UML 클래스 도해의 메타데이터

정보 정의에 바탕으로 정보를 가공하는 방법을 나타낸다.

3.1 UML 클래스 도해의 메타데이터 정보 저장

3.1.1 클래스의 메타데이터 정보 저장하기

클래스의 메타데이터 정보를 저장하는 것은 노드(다이아그램에서 클래스를 표현한 것)에 표시된 name, attribute, operation을 저장하는 것이며, 정보를 저장하는 순서는 아래의 그림 4와 같이 나타내었다. 먼저 선택된 노드를 식별한 다음 만약, 사용자가 입력한 name, attribute, operation 있다면 입력된 정보를 가지고 클래스정보저장소에 저장한다.

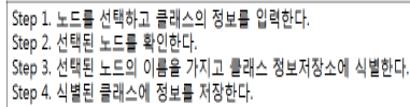


그림 4. 클래스의 메타데이터 정보를 저장하는 알고리즘

3.1.2 관계의 메타데이터 정보 저장하기

관계의 메타데이터 정보를 저장하는 것은 연결선(다이아그램에서 관계를 표현한 것)을 식별하여 from클래스, to클래스, 관계의 유형을 저장한다. 정보를 저장하는 순서는 그림 5와 같이 표현했다. 먼저, 시작노드를 식별하여 from클래스의 정보를 얻고, 도착노드를 식별하여 to클래스 정보를 얻는다. 마지막으로 노드와 노드 사이에 표현된 연결선의 유형을 식별하여 관계의 유형의 정보를 얻어 획득한 세 가지 정보를 관계정보저장소에 저장한다.

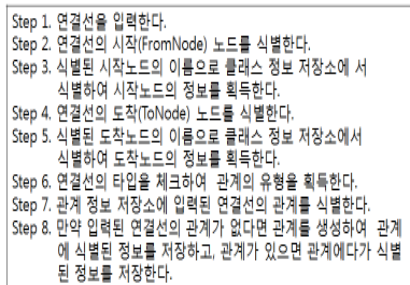


그림 5. 관계의 메타데이터 정보를 저장하는 알고리즘

3.2 UML 클래스 도해의 메타데이터 정보 불러오기

3.2.1 클래스의 메타데이터 정보 불러오기

클래스의 메타데이터 정보를 불러오기는 클래스정보를 필요로 하는 기능을 수행하기 위해 사용한다. 클래스 정보를 불러오는 순서는 그림 6과 같이 나타내었다. 클래스의 정보를 얻은 다음 그 기능을 수행한다.

Step 1. 선택된 노드의 이름을 식별한다.
 Step 2. 식별된 이름으로 클래스 정보 저장소에서 클래스를 식별한다.
 Step 3. 식별된 클래스에서 원하는 정보를 얻어온다

그림 6. 클래스의 메타데이터 정보를 불러오는 알고리즘

3.2.2 관계의 메타데이터 정보 불러오기

관계의 메타데이터 정보를 불러오기 또한, 관계의 정보를 필요로 하는 기능을 수행하기 위해 사용된다. 관계 정보를 불러오는 순서는 아래 그림 7과 같이 표현했다. 그리고 관계 정보를 획득한 다음 그 기능을 수행한다.

Step 1. 선택된 연결선의 시작노드를 식별한다.
 Step 2. 시작노드의 이름으로 클래스 정보 저장소에서 시작노드의 정보를 담고 있는 클래스를 식별한다.
 Step 3. 식별된 클래스에서 클래스 이름을 얻어온다.
 Step 4. 선택된 연결선의 도착노드를 식별한다.
 Step 5. 선택된 연결선의 타입을 식별한다.
 Step 6. 도착노드의 이름으로 클래스 정보 저장소에서 도착노드의 정보를 담고 있는 클래스를 식별한다.
 Step 7. 식별된 클래스에서 클래스 이름을 얻어온다.
 Step 8. 시작 클래스의 이름과 도착 클래스의 이름과 연결된 연결선의 타입의 정보를 조합하여 관계의 이름으로 설정한 다음 관계 정보 저장소에서 관계를 식별한다.
 Step 9. 식별된 관계에서 원하는 정보를 얻어온다.

그림 7. 관계의 메타데이터 정보를 불러오는 알고리즘

IV. 결 론

본 논문에서는 UML 클래스 도해의 처리도구를 위한 메타데이터의 정보 구축을 제안했다. 클래스 다이어그램에서 표현되는 클래스와 관계의 정보를 구축하였고 구축한 정보를 가지고 처리하는 메커니즘을 정의 하였다. 향후 과제로는 UML 클래스 도해의 처리 도구에서 표현하는 다이어그램을 코드 제너레이션 하여 코드로 변환시키는 기능을 지원하도록 하며, 이 기능을 수행하기 위한 메커니즘 설계에 대한 연구가 필요하다.

참고문헌

- [1] OMG Group, UML Sepcification, www.omg.org.
- [2] Grady Booch, Jim Rumbaugh, and Ivar jacobson, The Unified Modeling Language User Guide Second Edition, Addison Wesley,
- [3] Martin Fowler, UML Distilled Third Edition, Addison Wesley, 2004