

메트릭 기술 언어 : MDL4UML[†]

김태연*^o 박진욱** 채흥석*
 부산대학교 컴퓨터 공학과*
 {tykim, hschae}@pusan.ac.kr*
 LG전자**
 jinuki79@lge.com**

Metrics Description Language : MDL4UML

TaeYeon Kim*^o Jin-Uk Park**, Heung Seok Chae*
 Dept. of Computer Science and Engineering, Pusan National University *
 LG Electronics**

요약

UML을 이용한 소프트웨어의 설계가 널리 이용되고 있으므로 개발단계 초기의 소프트웨어 산출물인 UML을 대상으로 품질을 예측하는 메트릭의 개발 및 활용요구에 대응하기 위하여 메트릭 기술 언어를 개발 하였다. MDL4UML은 UML 모델에 적용 가능한 메트릭을 기존 OCL을 이용하여 기술하였을 경우 메트릭의 표현이 복잡하고 측정 대상 UML모델의 깊은 이해가 선행되어야 하는 점을 보완하기 위하여 연구된 메트릭 기술 언어이다. 본 연구에서는 OCL을 이용한 메트릭 기술의 난해함을 보완하기 위한 수단인 MDL4UML을 정의하고 이를 지원하기 위한 도구로 METOOSE를 개발하였다.

1. 서론

UML을 이용한 소프트웨어 설계가 폭넓게 이용되고 있다. 객체지향 모델의 제약을 표현하는 OCL의 용도를 확장하여 UML 모델에 적용할 메트릭을 기술하는 언어로 사용하는 연구가 다양하게 진행되었다[1]. OCL을 메트릭 기술 언어로 사용하는 방법은 OCL의 다양한 연산을 활용하여 UML 모델로부터 필요로 하는 정보를 얻는 방법이다.

그러나 OCL은 처음부터 메트릭을 기술하기 위해 만들어진 언어가 아니므로 OCL로 메트릭을 기술하기 위해서는 OCL과 UML 모델에 대한 높은 이해가 필요하며 OCL을 이용하여 메트릭을 표현하였을 경우 복잡한 OCL 문장으로 인하여 메트릭의 의미를 이해하는 데에 많은 어려움이 있었다. 특히, 다양한 UML 다이어그램을 이용하여 표현하는 메트릭에 대한 기술이 어려웠다.

따라서, 본 연구에서는 OCL의 기본 요소를 추상화시킨 새로운 메트릭 기술 언어(MDL4UML)를 정의하였다. MDL4UML은 복잡한 UML 모델 간의 연결을 자동으로 결정해주므로 메트릭을 기술하는 메트릭 디자이너는 복잡한 UML 모델에 관한 깊은 이해가 없이도 UML 다이어그램에 적용 가능한 메트릭을 쉽게 작성할 수 있다. 또, OCL의 기본 요소를 추상화함으로써 OCL을 이용하여 메트릭을 기술하는 메트릭 디자이너는 이해하기 쉽고 간략하게 원하는 메트릭을 기술할 수 있는 장점이 있다.

2. OCL을 이용한 메트릭 표현

본 절에서는 OCL을 이용하여 UML 모델에 적용 가능한 객체지향 메트릭을 표현하는 예를 보인다. 측정의 대상으로 그림 1의 클래스 다이어그램을 사용하여 대상 모델의 Cyclomatic Complexity (CC)를 측정해 본다[2]. CC는 프로그램의 복잡도를 측정하는 메트릭이다. 그림 2는 CC 메트릭 측정시에 클래스 다이어그램과 함께 사용될 액티비티 다이어그램이다.

CC 메트릭을 OCL로 표현하여 대상 다이어그램에 적용한 결과는 8이다. 예제 액티비티 다이어그램에서 CC가 8임을 확인할 수 있다. CC 메트릭을 OCL로 표현해 본 결과 OCL 표현만으로 메트릭의 내용을 이해하기가 난해하며 UML모델에 대한 높은 이해력을 요구하고 있다.

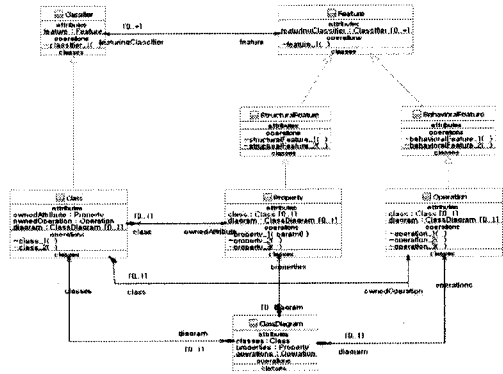


그림 1 예제 클래스 다이어그램

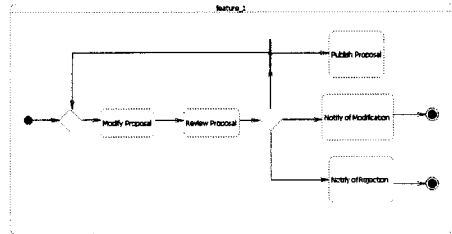


그림 2 예제 액티비티 다이어그램

```
self.packagedElement->select(p:PackageableElement| p.ocllsType
eOf(Class)=true).oclAsType(Class).ownedOperation->iterate(o:O
peration;cc:Integer=0| cc+Activity.allInstances()->select(a:Activi
y| a.name=o.name)->iterate(a;oc:Integer=0| oc+a.edge->size()-
a.node->select(ocllsKindOf(Action)=true)->size()+ 2))
```

3. MDL4UML

MDL4UML은 METRIC과 SUBMETRIC으로 구성되어 있으며 다음과 같 은 표현식으로 기술한다.

[†] "본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (ITA-2007-(C1090-0701-0032))

METRIC (메트릭이름) **FOR** (측정대상이름)[조건]
TYPE (타입) **VALUE** (value expression)

SUBMETRIC (메트릭이름) **FOR** (측정대상이름)[조건]
TYPE (타입) **VALUE** (value expression)

- 메트릭이름: 문자열로 표현되는 메트릭의 이름을 기술
 - 측정대상이름: UML모델 중 대상이 되는 요소의 이름
 - 조건: 측정을 원하는 대상물의 검색 조건
 - 타입: 메트릭 측정 결과 형태
 - value expression: 컨스탄트, OCL표현식, 집합항수
- MDL4UML의 수행 순서는 먼저 **SUBMETRIC**에 정의된 메트릭 결과값을 측정하고 그 결과를 상위 **METRIC**에서 이용하여 최종 측정값을 계산한다. MDL4UML의 이해를 돕기 위하여 MDL4UML를 이용하여 프로그램의 복잡도를 측정하는 CC메트릭을 기술하여 본다.

```

1 METRIC CC FOR Class TYPE Integer
2 VALUE SUM(MW)
3 BEGIN
4 SUBMETRIC MW FOR Operation TYPE Integer
5 VALUE SUM(AC)
6 BEGIN
7 SUBMETRIC AC FOR Activity[outer.name=name] TYPE Integer
8 VALUE edge->size()- node->select(ocl::isKindOf(Action)=true)->size()+2
9 END
10 END
    
```

6번 라인에서 9번 라인에 SUBMETRIC이 중첩되어 사용된 것을 확인할 수 있다. 7번 라인의 outer조건은 MDL4UML에서 정의한 요소로 상위 측정대상과의 관계를 자동으로 찾아주게 된다.

MDL4UML은 메트릭을 기술함에 있어 메트릭 기술 언어로서의 OCL과 비교하여 보았을 때 다음과 같은 특징을 가진다.

- SUBMETRIC을 이용한 모듈화
 복잡한 메트릭을 기술하는 경우에 SUBMETRIC을 이용하여 메트릭에 필요한 데이터를 분할하여 처리할 수 있다.
- OCL의 select() 표현식 사용 빈도 감소
 UML 대상간의 연결에 OCL의 select() 오퍼레이션을 사용함으로써 OCL 표현이 길어지고 이해도가 떨어졌다. MDL4UML은 UML 대상간의 연결을 자동으로 찾아주므로 select() 오퍼레이션의 사용 빈도를 감소시킨다.
- 집합항수의 지원
 OCL을 이용한 메트릭의 기술사 iterate() 오퍼레이션을 이용하여 대상 집합의 value expression을 합산하거나 개수를 구하였으나 MDL4UML은 집합항수를 이용하여 대상집합에서 원하는 결과를 얻을 수 있다.

OCL로 기술한 경우와 비교해 보았을 때 측정대상 간에 연결을 위한 select() 오퍼레이션이 전혀 없으며 Operation 집합을 대상으로 COUNT(OP) 집합항수를 사용함으로써 iterate() 구문 역시 나타나지 않는다. CC 메트릭을 MDL4UML로 기술한 예는 서로 다른 대상의 SUBMETRIC을 사용하는 유형에서 설명하였다. MDL4UML을 이용하여 복잡한 메트릭을 기술하는 경우에 SUBMETRIC을 이용하여 분할하여 기술할 수 있다. CC 메트릭의 경우 먼저 클래스 내의 메소드들의 이름과 일치하는 액티비티 다이어그램의 엣지와 노드를 이용하여 CC 값을 구하고 다음으로 메소드별로 결과값을 합산하여 전체 프로그램의 CC 값을 구하고 있다.

또, 측정 대상간의 연결이 자동으로 이루어지므로 OCL 표현에서 반복적으로 나타나는 select() 오퍼레이션의 뚜렷한 감소를 확인할 수 있

으며 클래스 다이어그램과 액티비티 다이어그램을 이용하여 기술된 CC 메트릭의 크기 역시 24% 감소하였다.

4. 자동화 지원 도구

MDL4UML로 기술된 메트릭을 UML모델에 실질적으로 적용해 보기 위하여 소프트웨어 측정 도구인 METOOSE를 구현하였다. METOOSE는 MDL4UML을 OCL로 변환할 수 있고 UML 모델을 대상으로 메트릭 결과를 측정하고 관리할 수 있다. METOOSE는 두개의 중요한 도구로 구성되어 있다. MDL4UML을 입력 받아 OCL 표현으로 변환해 주는 MDL2OCL과 OCL을 UML 모델에 적용하여 결과를 측정하고 관리하는 MetriUs이다. 그림3 은 METOOSE의 아키텍처를 도식화 한 그림이다.

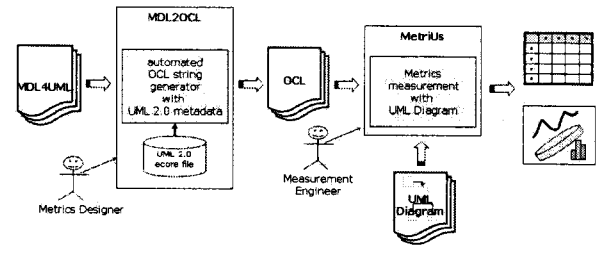


그림 3 METOOSE의 아키텍처

5. 관련 연구

[3]의 연구는 UML 모델을 대상으로 메트릭 적용 프레임워크를 설계하고 메트릭 기술 언어 및 기본 라이브러리를 구현하였다. [4]에서는 XML파일 형식으로 메트릭을 기술한 후 SDMetrics 도구를 이용하여 메트릭을 측정할 수 있다. [5]의 연구는 객체지향 디자인 모델(ODEM)을 제시하고 정형 인식자를 사용하여 메트릭을 기술하였다. 그러나 세 연구 모두 도구 및 라이브러리를 공개하고 있지 않다.

6. 결론

OCL을 이용하여 UML 모델의 메트릭을 기술하는 경우에 UML 모델에 관한 높은 이해력이 필요하였고 OCL 표현의 복잡성을 유발하였다. 메트릭을 기술하기 위한 언어인 MDL4UML을 이용하면 UML 모델간의 복잡한 연결관계를 자동으로 찾아서 OCL 표현으로 변경할 수 있으며 OCL 표현을 추상화하여 메트릭을 기술함으로써 메트릭 디자이너는 이해하기 쉽고 간편하게 메트릭을 작성할 수 있을 것으로 예상된다.

참고 문헌

- [1] J.A McQuillan and J. F. Power. "Experiences of using the Dagstuhl Middle Metamodel for defining software metrics", Proceeding of the 4th International Symposium on Principles and Practice of Programming in Java, pp. 194-198, 2006.
- [2] McCabe, Thomas J. & Watson, Arthur H. "Software Complexity." CrossTalk: The Journal of Defense Software Engineering, 1994.
- [3] Alikacem, E. H., Sahroui, H. "Generic Metric Extraction Framework" In Proceedings of the 16th International Workshop on Software Measurement and Metrik Kongress, 2006.
- [4] SDMetrics. SDMetrics Metric Language. "http://www.sdmetrics.com/CustoMetrics.html".
- [5] R. Reibing. "Towards a model for object-oriented design measurement", International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering, 2001.