

# UML 기반 설계도구에서의 프로파일 지원을 위한 모델

김정일<sup>o</sup>

㈜플라스틱소프트웨어 소프트웨어 기술 연구소

jikim<sup>o</sup>@plasticsoftware.com

## Applicable Profile Model for UML-based Modeling Tool

Jeongil Kim<sup>o</sup>

Software Technology Laboratory, Plastic Software, Inc

### 요 약

UML은 범용의 목적으로 설계된 모델링 언어이지만 많은 도메인 영역에서 사용되고 있다. 그것은 UML을 각 도메인의 특성에 맞게 확장할 수 있는 프로파일이라는 메커니즘을 자체적으로 보유하고 있기 때문이다. UML 표준 명세에 기술되어 있는 확장 메커니즘과 프로파일은 그 자체로 매우 잘 설계된 모델이나 그대로 설계 도구에 도입하기에는 부적합한 면이 많다. 본 논문에서는 UML 기반의 설계 도구에서 프로파일을 수용할 수 있도록 확장 메커니즘 모델을 개선하고 프로파일을 정의할 수 있는 정형적인 방법들을 제시한다.

### 1. 서 론

1997년 OMG에 의해서 UML 1.1이 발표된 이후로 UML은 꾸준히 발전을 거듭해 왔으며, 새로운 버전이 발표될 때마다 다양한 개념들을 수용해 왔다. 현재 UML은 소프트웨어 모델의 설계 시에 없어서는 안 될 필수적인 표준 모델링 언어로 자리 잡았으며, 수 많은 툴 벤더들에 의해 지원되고 있다.

UML은 소프트웨어 모델 뿐만 아니라 데이터베이스 모델링, 비즈니스 프로세스 모델링, 웹 애플리케이션 모델링 등 다양한 분야에 적용할 수 있다. 이것은 UML이 특정 도메인에 종속적이지 않은 범용의 목적으로 설계되었을 뿐만 아니라, 각 도메인 영역에서 사용되어지는 다양한 표현들 수용할 수 있는 빌트-인 확장 메커니즘(Built-in Extension Mechanism)을 자체적으로 구비하고 있기 때문이다. 특히 프로파일은 최근 새로운 개발 방법론으로 주목을 받고 있는 MDA(Model Driven Architecture)[1]에서 MOF(Meta Object Facility)[2]와 함께 모델링을 지원하기 위한 필수적인 기능으로 그 중요성이 더욱 높아지고 있다.

지금까지 전 세계적으로 수 많은 툴 벤더들이 UML을 기반으로 하는 소프트웨어 모델링 도구를 제작하였으며, 그 중 일부는 UML 확장 메커니즘과 프로파일을 도구 상에 도입하여 운용하고 있다. 그러나 UML 표준 명세에 정의되어 있는 프로파일 모델은 그 운용방법을 명확하게 제시하고 있지 않기 때문에 모델링 도구를 마다 구현이 제각기 다르다.

본 논문에서는 UML 기반의 모델링 도구에서 프로파일 개념을 도입할 때 고려해야 할 점들과 그 개선 방안을 기술한다. 그 결과로 UML 표준 명세에 명시된 확장 메커니즘의 메타 모델을 모델링 도구에서 수용하기 위한 현실적인 모델을 제시하고, 프로파일의 정의와 편집을 효과적으로 할 수 있는 정형적인 방법들을 열거한다.

### 2. 연구 배경

#### 2.1 UML 확장 메커니즘과 프로파일

프로파일은 UML에서 사용할 수 있는 확장 메커니즘의 하나로 MOF와 비교하여 경량의 확장 메커니즘(light-weighted extension mechanism)[1]이라고 불린다. UML 표준 명세에서 설명하고 있는 프로파일과 확장 메커니즘의 요소들의 의미는 다음과 같다.

- 프로파일(Profile) : 프로파일은 <<profile>> 스테레오타입의 패키지(Package) 요소이다. 프로파일은 특정 도메인을 기술하거나 메타모델을 확장하기 위한 스테레오타입, 태그정의, 제약사항 등을 포함하고 있다
- 스테레오타입(Stereotype) : 스테레오타입은 기존의 메타클래스에 추가적인 속성값과 제약사항, 그리고 시각적인 표기법을 정의한 하나의 모델 요소이다. 스테레오타입은 UML이 정의하고 있는 메타모델 클래스 계층에 추가적인 분류 메커니즘을 제공한다. 따라서 스테레오타입의 이름은 UML 메타모델에 정의되어 있는 각 메타클래스의 이름과 달라야 하는 제약이 있다.
- 태그정의(Tag Definitions) : 태그정의는 모델 요소에 추가될 수 있는 새로운 속성을 정의한다. 태그정의에 따라 각각 모델 요소에 명시되는 실제 속성값은 태그값(Tagged Values)를 사용하여 저장한다.
- 제약사항(Constraints) : 제약사항은 모델 요소나 스테레오타입에 추가되어 그 의미를 정제하는데 사용된다.



```

<!ELEMENT BASECLASSES (BASECLASS+)>
<!ELEMENT BASECLASS (#PCDATA)>
<!ELEMENT PARENT (#PCDATA)>
<!ELEMENT TAGDEFINITIONSET (#PCDATA)>
<!ELEMENT ICON (#PCDATA)>
<!ATTLIST ICON
    minWidth CDATA #IMPLIED
    minHeight CDATA #IMPLIED
>
<!ELEMENT TAGDEFINITIONSETLIST (TAGDEFINITIONSET*)>
<!ELEMENT TAGDEFINITIONSET (NAME, BASECLASSES, TAGDEFINITIONLIST)>
<!ELEMENT TAGDEFINITIONLIST (TAGDEFINITION+)>
<!ELEMENT TAGDEFINITION (NAME, TAGTYPE, DEFAULTDATAVALUE?, LITERALS?>
<!ATTLIST TAGDEFINITION
    lock CDATA #IMPLIED
>
<!ELEMENT TAGTYPE (#PCDATA)>
<!ATTLIST TAGTYPE
    referenceType CDATA #IMPLIED
>
<!ELEMENT DEFAULTDATAVALUE (#PCDATA)>
<!ELEMENT LITERALS (LITERAL+)>
<!ELEMENT LITERAL (#PCDATA)>
<!ELEMENT DATATYPELIST (DATATYPE*)>
<!ELEMENT DATATYPE (NAME)>
    
```

(그림 2) 프로파일 정의의 문서 DTD

프로파일 정의의 문서는 크게 프로파일 자체의 기본적인 정보를 기술하는 헤더 부분과 프로파일이 정의하는 UML 확장 메커니즘을 기술하는 바디 부분으로 구성된다.

문서의 헤더는 문서를 식별할 수 있는 프로파일의 이름과 문서의 버전, 설명등으로 구성된다. 프로파일의 이름은 다른 프로파일 문서와 구별되는 유일한 아이디(ID) 역할을 하는 것으로 이름의 중복에 따른 충돌을 방지하기 위하여 프로파일을 정의한 저작자를 표시하는 네임스페이스를 부여한다.

문서의 바디 부분에는 프로파일 패키지를 구성하는 스테레오타입(Stereotype), 태그정의 집합(TagDefinitionSet), 태그정의(TagDefinition), 데이터타입(DataType) 등을 기술한다. 프로그램은 초기화 수행 시에 시스템 레지스트리에서 시스템에 설치된 프로파일의 등록 정보를 참조하여 프로파일 정의 문서를 읽어 들인 다음 프로파일과 확장 메커니즘을 위한 객체 구조를 생성한다.

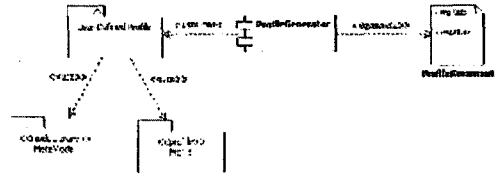
#### 4.2 프로파일 모델링(Profile Modeling)

앞서 제시한 프로파일 정의 문서가 비록 잘 정의된 XML 형식의 문서이기는 하지만 일련의 태그로 나열된 텍스트이다. 만약 모델링 도구에서 프로파일을 시각적으로 편집가능한 기능을 제공한다면 프로파일의 효율성이 더 증대될 것이다.

UML 표준 명세에는 프로파일 편집을 위한 구체적인 방법이 제시되어 있지 않지만 OMG의 다른 문서에서 프로파일을 시각적으로 편집하기 위한 좋은 방법을 찾아 볼 수 있다.

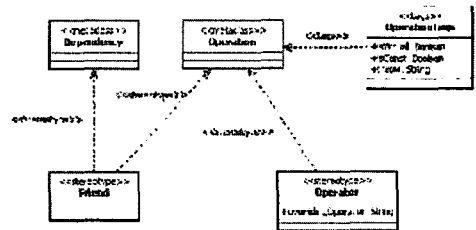
OMG는 CORBA 프로파일 명세에서 프로파일 편집을 위하여 가상 메타모델링(Virtual MetaModeling)[4]이라는 새로운 기법을 소개하였다. 이것은 MOF에서의 M1 수준의 요소를 사용하여 M2 수준의 메타클래스와 메타속성 등을 표현하려는 것으로 프로파일을 정의하는데 매우 효과적인 방법이다. 이 기법을 응용하면 UML 표기법을 사용하여 프로파일을 모델링 하고 이로부터 프로파일 정의 문서를 쉽게 생성할 수 있다.

(그림 3)은 프로파일을 모델링하여 프로파일 정의 문서를 생성하는 일련의 과정을 도식화 한 것이다.



(그림 3) 프로파일 모델링 개념도

프로파일 저작자는 가상 메타모델링 기법으로 프로파일 모델을 설계하고 프로파일 문서 생성기를 사용하여 프로파일 정의 문서를 생성한다. 이를 위해 모델링 도구는 UML 메타모델의 계층적 구조를 포함하는 모델 라이브러리와 프로파일을 정의하기 위한 프로파일(Profile for Profile)을 제공하도록 한다. (그림 4)는 가상 메타모델링 기법을 사용하여 설계한 C++ 프로파일의 일부를 예로 보인 것이다.



(그림 4) C++ 프로파일의 일부

#### 5. 결론 및 향후 과제

본 논문에서는 UML 프로파일을 설계 도구에서 수용하기 위한 현실적인 방안을 제시하였다. 그 결과로 확장 메커니즘 모델에 대한 약간의 변화를 통해서 프로파일 개념의 효과적인 운용 방법을 얻을 수 있었다. 향후 과제로 최근 표준에 관한 제정이 활발하게 진행되고 있는 UML 2.0에서 정의하고 있는 프로파일에 대한 연구도 이루어져야 할 것이다.

#### 6. 참고 문헌

- [1] David S. Frankel, "Model Driven Architecture" OMG Press, 2003.
- [2] Object Management Group, "Meta Object Facility(MOF) Specification version 1.4" April 2002.
- [3] Object Management Group, "UML Specification Version 1.4" April 2002.
- [4] Object Management Group, "UML Profile for CORBA, Version 1.1" January 2001.