ClubUML

CSYE7945 Spring 2013

The Meta Model of Class Diagram

Di Huang

02-20-2013

Revision: 1.1

Table of Contents

Introduction 3

Top level of Meta Model 4

Lower Level of Meta Model 5

Operation 5

Association 6

Property 7

Conclusion 7

Appendix 8

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author(s)** | **Notes** |
| 1.0 | 2/20/13 | Di Huang | Initial version |
| 1.1 | 3/15/13 | Richard Do | Removed constraints from Operation diagram |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

This document describes the Class Diagram meta model which is a subset of the OMG UML super structure specification that is too complex for our project. We can’t follow every detailed guide in that document to create a tool to modify UML diagram. However, that is still our most important reference resource.

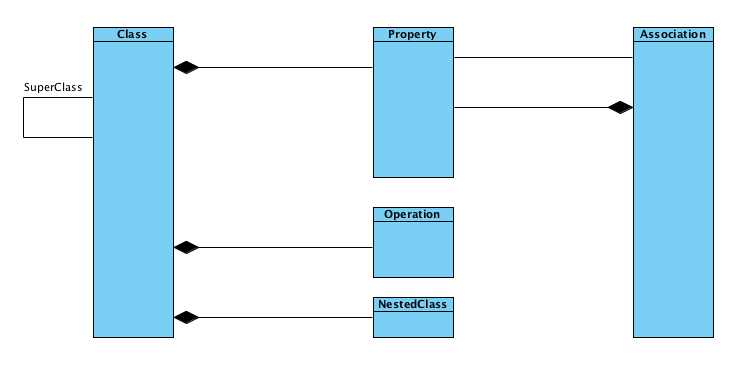
In order to achieve our goal, we created a simpler, easier to understand meta model which references the OMG UML super structure specification.

# Top level of Meta Model

When we saw a class diagram, what is the most obvious thing? The Classes and associations. The following diagram is the top level of Meta Model of Class diagram.

We can see that:

1. Class is compromised by properties, operations, and sometime, nestedClass.
2. Property contains 2 type of properties:
   1. The property of an attribute
   2. The property of an end of a association
3. Classes are associated by associations, and which may imply that these classes contain related properties.
4. Operation is a kind of action feature which is different type of diagram. In order to simplify, we only concern that visible parts. See detail in next section.
5. Nest Class is a kind of special class which is nested in another class.



# Lower Level of Meta Model

## Operation

## 

Operation contains 2 types of elements:

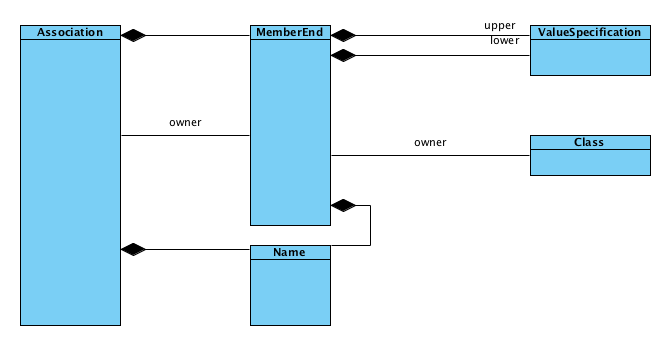
1. Parameter, that is a specification of an argument used to pass information into or out of an invocation of a behavioral feature, that is, an operation
2. Type, that constrains a type represented by a type element. In our project, the return type, and the parameter type can impact one operation

## Association

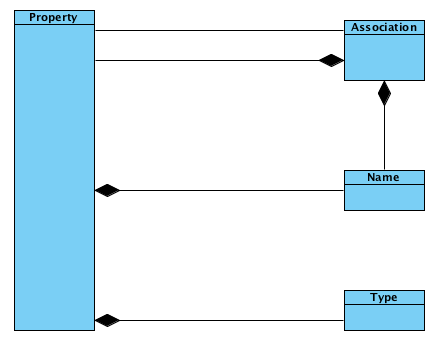
An association describes a set of tuples whose values refer to typed instances. That means it describe a kind of relationship between at least 2 ends. Association also contain two types:

1. One ordered type, about this type, one end owns one property that represents this attribute. It is a kind of navigable association.
2. One normal type, about this type, no ends owns property. The association itself owns two ends. This is kind of association represented by association itself, classifier is not owner of this association.

No matter what type, all type contains a name, and multiplicity.

A multiplicity can be separated as a upper value and lower value. See following diagram. 

## Property



A property related to a classifier by ownedAttribute represents an attribute, and it may also represent an association end. It relates an instance of the class to a value or collection of values of the type of the attribute.

A property related to an Association by memberEnd or its specializations represents an end of the association. The type of property is the type of the end of the association.

# Conclusion

After dived into the super structure of UML diagram, I found that I had a lot of misunderstanding about class diagram before. For example, an association sometimes is owned by a class, sometimes it owned by itself. In order to make sure that our UML tool is correct. It’s really necessary to get familiar with the super structure of UML diagram.

# Appendix

*Unified Modeling Language: Superstructure* version 2.0 formal/05-07-04