Considerations for UML/Magicdraw associations and modular models.

In modeling threat/risk, other conceptual models and any substantial architecture we need a consistent and usable approach for modularization. This is impacted by UML structural considerations.

The issue

Any substantial model should be modularized and allow for independent ownership and evolution of modules. We also want to be able to utilize standards based model modules and build on them. Both of these requirements imply that:

1. We will have “read only” models containing packages and classes we want to use
2. We will want to make associations with these classes.
3. We want consistency of usage, semantics and representation regardless of location of classes.

A simple model module is below in “Model1”, we can define any kind of class and association. Of interest is the “ownership” of the end, an odd concept added in UML-2. If I have an end owned by the class at the end, certain features are enabled. Some are tooling features, others are native to UML.

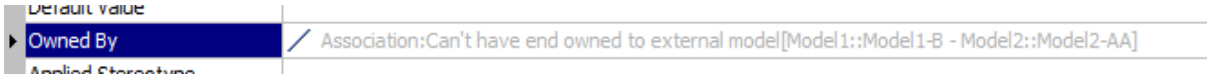
Features of class-owned-and:

- I can see the end as a “part” in a composite (See class A composite)
- I can navigate in expressions (OWL and ALF)
- I can see the end as part of the class doc (tooling)

Associations across read-only models

I can have an association between classes In all the UML tools I know of, and MD as shown here, an association to a read-only class can’t have that end as association owned.

Note “Model2” which uses that model as read-only. MD requires that any association to model1 classes is “association owned”.



Question 1: Is this a requirement of MOF? It seems to be, but Pete has said otherwise. However, it is the way it is in current tools, so I think we must live with it.

So: We must be able to use association owned ends. If we want to be consistent, it would seem to make sense that all ends should be association owned. Thus, this “owned end” idea is bad, but again, we have to live with it or wait 4 years for it to change in UML and tools.

So what do we have to do?

- We have to always require association owned ands
- We must not use composition “parts” as SIMF and threat/risk are doing
- We have to adjust tooling to document and otherwise treat association owned ends like they are currently doing for class owned ands.

Work around

There is a work-around. Never make associations to external classes, subtype (or “equivalent” them instead. However, this introduces a lot of complexity.

Or, do I have this all wrong???

Model1

Diagram: Model1

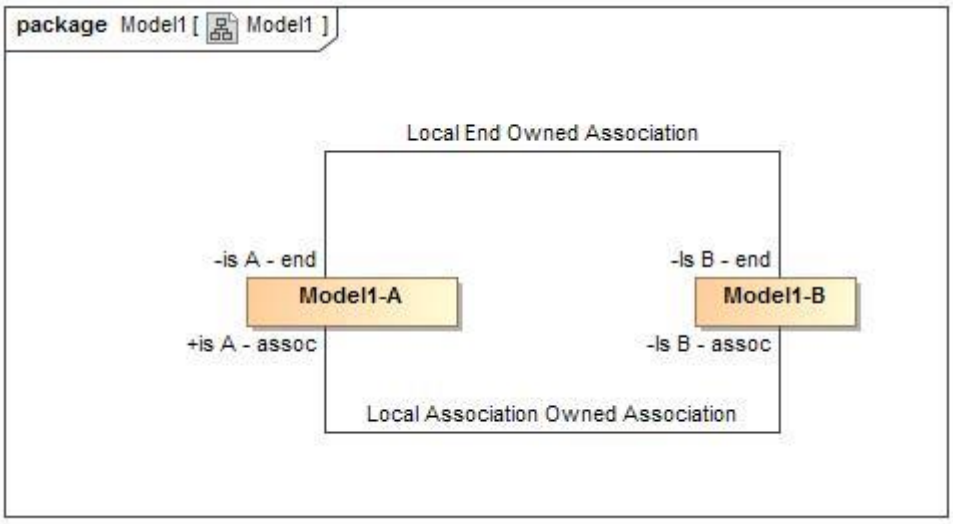


Figure 1. Model1

Association Local Association Owned Association

package Model1

Association Ends

- Is B - assoc : Model1-B
- is A - assoc : Model1-A

Association Local End Owned Association

package Model1

Association Ends

- Is B - end : Model1-B
- is A - end : Model1-A

Class Model1-A

package Model1

Associations

Is B - end : Model1-B

Class Model1-B

package Model1

Associations

is A - end : Model1-A

Model2

Diagram: Model2

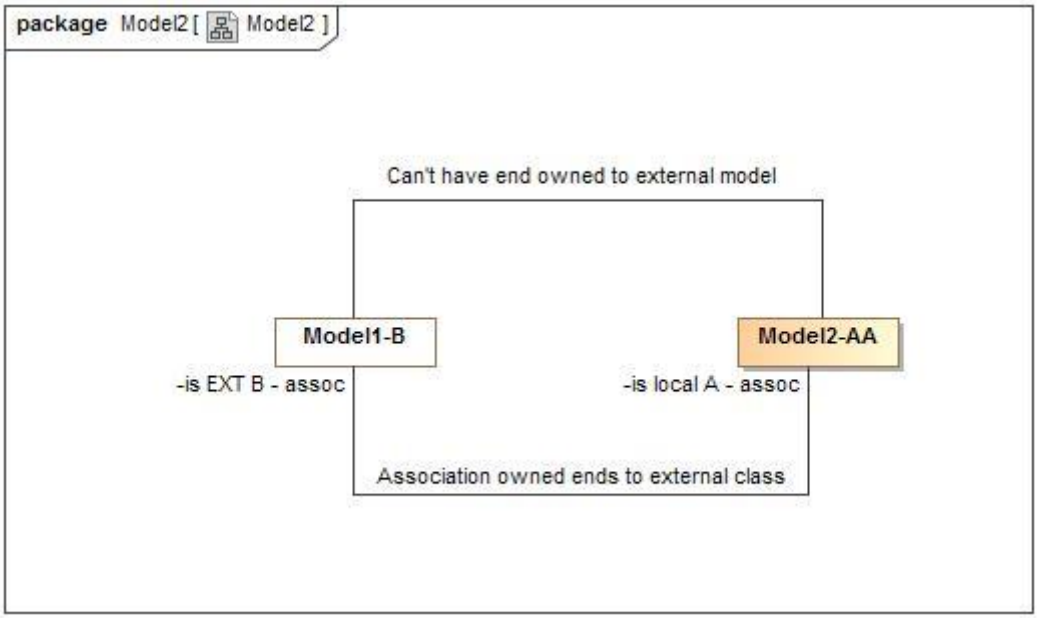


Figure 2. Model2

Class A Composite

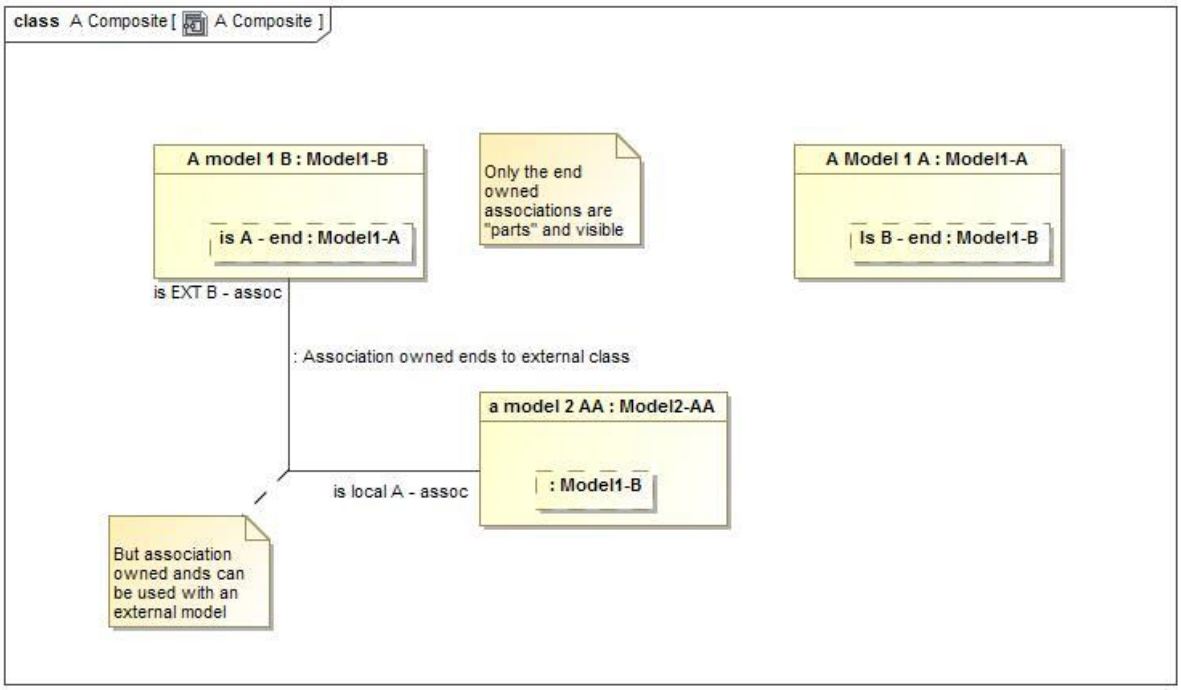


Figure 1. A Composite

package Model2

Attributes

- a model 2 AA : Model2-AA
- A model 1 B : Model1-B
- A Model 1 A : Model1-A

Class Model2-AA

package Model2

Associations

- : Model1-B