# Discussion of the “rigidity” of “roles”, “phases” and “facets”

Cory Casanave 5/26/2016

## Overview

OntoUML and SIMF intentionally have a great deal in common and there is a mutual desire for convergence. Due to varieties in target audience, in some cases OntoUML may provide additional rigor that may be layered on SIMF, where SIMF allows for less formality.

SIMF and OntoUML define roles and phases for the same reasons and with the same intent. Our shared opinion is that these concepts are crucial for properly representing a conceptual model and that they must be distinguished from the persistent identity of an entity (OntoUML “Kind”).

The difference is with assumptions of “rigidity” and the differentiation made in OntoUML between “Anti-Rigid Sortal types” (e.g. “role”) and “Anti Rigid Mixin Types” (e.g. Role Mixin”). SIMF does not make these distinctions. In SIMF types are either rigid (e.g. “Person”) or non rigid (e.g. Role). Non rigid types are a “Facet” (Role and phase are subtypes of Facet).

Our intent in this document is to make initiate a discussion leading to choices as to the best SIMF conceptual framework with a well-defined relationship with OntoUML.

This, of course, reflects my possibly inaccurate interpretation of Onto UML.

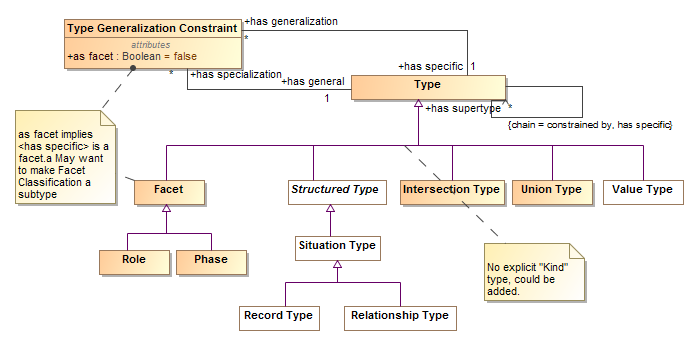
## Onto UML meta model fragment:



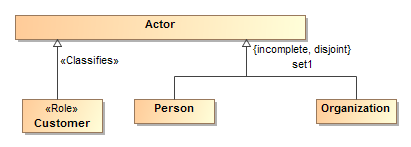
## Onto UML example:



## The SIMF meta model



## SIMF Example



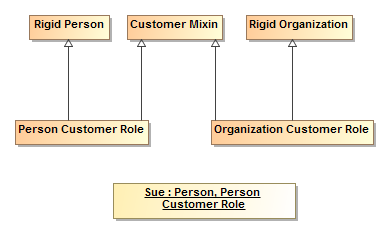


In SIMF the rigid identity comes from Sue as a person, Customer is a classification of Sue (perhaps in a very specific context). Customer must classify an instance that has a rigid type compatible with what it <<classifies>> so the identity is asserted at the instance level. A facet type does not assert a rigid identity for its instances.

Note that many implementation frameworks would make the role a different technical object which would allow for the SIMF flexibility of any assertion being in any context.

Note that SIMF does not prevent a role as a supertype of a rigid type, which would be a role in OntoUML. In SIMF this would be a required role of a rigid type.

## OntoUML Example with instance



In Onto UML rigid identity still comes from Person – Identity of a role is identified in the role type. A Mixin type may not directly classify a rigid type (is this correct?).

## Concerns with OntoUML approach

While we would like convergence with OntoUML, we have concerns with the approach – both practical and logical.

### Rigid/Anti-rigid conflict?

In the OntoUML example, above – Sue is both an anti-rigid “Person Customer Role” and a rigid “Person”. How can Sue be rigid and anti-rigid at the same time?

That Sue gets “Personness” from 2 sources seems redundant.

### Correspondence with user concepts

A primary goal of SIMF is to retain as close a correspondence as possible with how stakeholders conceive of their domain while providing semantic clarity and precision. Introduction of extra concepts and terms that do not have this correspondence should be minimized. Anything beyond what they conceive will be, to them, model “noise”. Noisy models are hard to understand.

The two “flavors” of roles – as a mixin and anti-rigid-sortal does not seem to have a correspondence with user conceptualization. It may be difficult to explain to the model author as well as to stakeholders viewing the model.

To be honest, the distinction between “mixin” and “non-rigid” or why a mixin is not a sortal is not that clear to me (Cory).

### Model complexity

The addition of a subtype of a mixin for every possible concrete subtype of the base of the mixin results in a set of types that is the number of role mixins \* the number of concrete types to which they may apply. This could have a dramatic impact on the number of classes and the job of determining which classes to use for a particular instance. For example, in threat/risk – this could be on the order of 1500 (a guess) where there are currently about 500 classes. It would be easy to forget to define some of these types, which would be an error.

### Return on investment

I think there is agreement that the level of semantic precision has to match the needs. There should be some “return” for asking the user to make some distinction or assertion. We don’t see how the “anti-rigid” (vs. non rigid) distinction and additional complexity is providing sufficient value, at least for the federation use case. However, we don’t have experience with strict OntoUML models so may be missing something.

## Issues with the SIMF approach

Our understanding of the issue with the SIMF approach is that the kind of identity of a facet/role is not specified in the type hierarchy. An organization and a person may have different kinds of identity so the kind of identity the customer role has is unspecified.

In SIMF we are not seeing this as an issue as the rigid identity is provided by the instance, the form of that identity for the role type does not need to be explicit (however making it explicit in a subtype is not an issue). In addition, it is not clear that all rigid types (e.g. organizations) will have the same form of identity at all levels of abstraction – of course they must have some identity at the instance level.

There may also be some model validity checking that can be improved with the OntoUML approach.

SIMF requires a common subertype of all concrete classes the role <<classifies>>. This supertype may or may not correspond to a user relevant concept, so in some cases may also be “noise”. However introducing a supertype rather than subtypes introduces far less noise and in many cases becomes an reasonable target for relationships.

## So what to do?

A decision needs to be made for SIMF as well as possible future versions of OntoUML as to what to require or allow for mixins, non rigid sortals and SIMF facets. If not the same our goal should be to at least have the relationship between the concepts well defined. If they are different, it would be desirable for OntoUML to be cleanly layered on top of SIMF.

One potential would be to interpret a SIMF Facet as mapping to a pattern of a OntoUML Mixin and a non-rigid sortal for each concrete supertype of the type the Mixin <<classifies>>.

The other difference is that SIMF allows a concrete type to inherit a role as a required (rather than dynamic) type. This is an OntoUML error. This pattern has proved useful in threat/risk (e.g. an organization inherits the authority role). I don’t know about FIBO. Allowing this pattern is also a choice, we would prefer to continue to allow it. Initially we did not allow it but found it needed to easily express the domain intent.

We welcome other thoughts and discussion.