FIBO NoMagic Requirements for Business Presentation

*9 April 2014*

# Overview

This document sets out the specification for business-facing presentation of FIBO content in Cameo.

Requirements are aligned across two dimensions: interactivity (static diagrams versus interactive editing of model content in SME reviews), and content, covering the additional requirements for visual aliasing of complex model patterns for restrictions.

Most diagrams showing the requirements for modeling of complex structure should be regarded as provisional and may be improved upon during development of this work. Similarly colors are not a hard and fast requirements and are subject to ongoing conversations within the OMG, but the ability to set these and have them consistently rendered is a requirement.

An additional feature of the pre-existing diagrams, that of assigning “Archetypes” to classes and properties, is also covered here and is regarded as a later element of the work.

## Requirements Summary

The overall requirements include:

* Rendering of existing classes and properties and disjoints and inverses
* Business facing rendering of restrictions
  + Describable as refinements and/or reuse of properties
* Ability to update the model in real time
* Ability to view and edit definitions
* Ability to generate additional synonyms
* Ability to review and edit provenance metadata
* Creating spreadsheets of terms, definitions and synonyms and metadata, etc. reports for off line review
* Create clean diagram by drag and drop classes from old diagram to new diagram while maintaining relationships for PPT presentations

Contents

[Overview 1](#_Toc384897319)

[Requirements Summary 1](#_Toc384897320)

[Introduction 3](#_Toc384897321)

[Background 4](#_Toc384897322)

[References 4](#_Toc384897323)

[Basic Content Static View 5](#_Toc384897324)

[Extended Content Static View 7](#_Toc384897325)

[Aliasing Label Values 9](#_Toc384897326)

[Cascaded Restrictions: Discussion Point 11](#_Toc384897327)

[More Advanced Renditions 12](#_Toc384897328)

[Basic Content Interactive Functionality 16](#_Toc384897329)

[Details 17](#_Toc384897330)

[Extended Content Interactive Functionality 18](#_Toc384897331)

[Ideal behavior: 18](#_Toc384897332)

[Under the Hood 19](#_Toc384897333)

[Archetypes 19](#_Toc384897334)

# Introduction

Requirements are described along two axes as shown in Figure 1:

**Figure 1: Dimensions of the Business Facing Diagram problem**

**STATIC VIEW:**

OWL Restrictions

Rendition of one graphical combination with another

All features of one map to features of the other

**DRAG AND DROP:**

Classes, Properties (AssClass)

Interactive drag and drop of the business “alias” pattern for OWL Restriction.

**STATIC VIEW:**

Classes, Properties (AssClass)

Rendition of class and association elements WITHOUT:

Stereotypes, End Labels etc.

Pre-configured color

**DRAG AND DROP:**

Classes, Properties (AssClass)

Interactive drag and drop of Class and AssClass, Generalization etc.

**INTERACTIVITY**

**C O N T E N T**

# Background

One of the main principles behind FIBO is the provision of diagrams suitable for a business audience. This covers the production of static diagrams (“slideware”) for off-line circulation, and the activities involved in industry Subject Matter Expert (SME) reviews.

An existing set of FIBO models exists, called the “Semantics Repository”. This may be found at the following URL, where a copy of the Enterprise Architect diagram may be downloaded. This model content has been developed in conjunction with industry SMEs following a formal SME review process, which is now to be carried out using Cameo from NoMagic. The key thrust of these requirements is to be able to do in Cameo what was previously done in Enterprise Architect. An additional requirement is to extend the SME review functionality which was available in Enterprise Architect, to include the business-facing rendition of OWL Restrictions.

A separate document sets out a more thorough exploration of the possible ways to represent OWL restrictions, including the kinds of complex restrictions “cascades” which are a feature of the new OMG specification models. That document also analyses some of these cascades in order to establish whether the proposed visual notations shed any light on the business implications of those complex patterns.

## References

1. Business Presentation of Restrictions v02.docx from EDM Council (not published)

# Basic Content Static View

* Create clean diagram by drag and drop elements from Containment Tree to new diagram while maintaining relationships

This is essentially a support activity. NoMagic is believed to support all these requirements, however we have had difficult replicating all these.

What’s needed:

* Create a new “Diagram Type” based on UML Class Diagram
* Drag and drop existing elements from the Containment Tree
* On dropping a Class:
  + Color matches a value pre-selected for all classes of a given stereotype for this diagram type
    - There are 3 or more separate Class stereotypes (owlClass, unionClass etc.)
    - AssClass class elements must also be color coded for the diagram type against a given AssClass stereotype (there are two of these)
  + Stereotype text is not rendered
    - Cameo menu option: “Symbol(s) Properties/Show Stereotypes/Do not display”
    - Alternatively may use “Shape image”, if this is used in rendering the Diagram Type colors.
  + Partitions for Attributes and for Operations not shown
    - Cameo menu option “Symbol(s) Properties/Suppress Attributes=True”
    - Cameo menu option “Symbol(s) Properties/Suppress Operations=True”
* On dropping a class which has an existing relationship to another class
  + The Association Class or Dependency is drawn on the diagram (rather than having to explore via “show related element”); that is, change the default for this behavior.
  + These are drawn subject to the rules below.
* On dropping or automatically inserting an Association Class
  + Color matches a value pre-selected for all AssClass class elements and association elements of a given stereotype for this diagram type
    - AssClass class elements must also be color coded for the diagram type against a given AssClass stereotype (there are two of these)
      * The line color is a darker rendition of the class color and both musts be settable (e.g. Fill Color v Outline Color in some products)
  + Stereotype text is not rendered
  + Association end labels are not rendered
    - This includes the association end name, copies of the association name with a + sign, and cardinalities
    - At present, even when these are deleted from a diagram in Cameo, reloading the diagram causes a copy of the association name with a + sign to appear as an association end name, and a default multiplicity element \* both to appear.
    - We’d like that fixed for the “technical” diagrams as well (these are beyond the scope of this specification).
* On automatically inserting a dependency
  + The stereotype is not shown
  + The name (if there is one) is shown; or one may be added with minimal clicks.
  + A chosen default color for that stereotype is shown
    - If it is not possible to associate a stereotype with a color, then a pre-selected color, probably red, is consistently rendered.

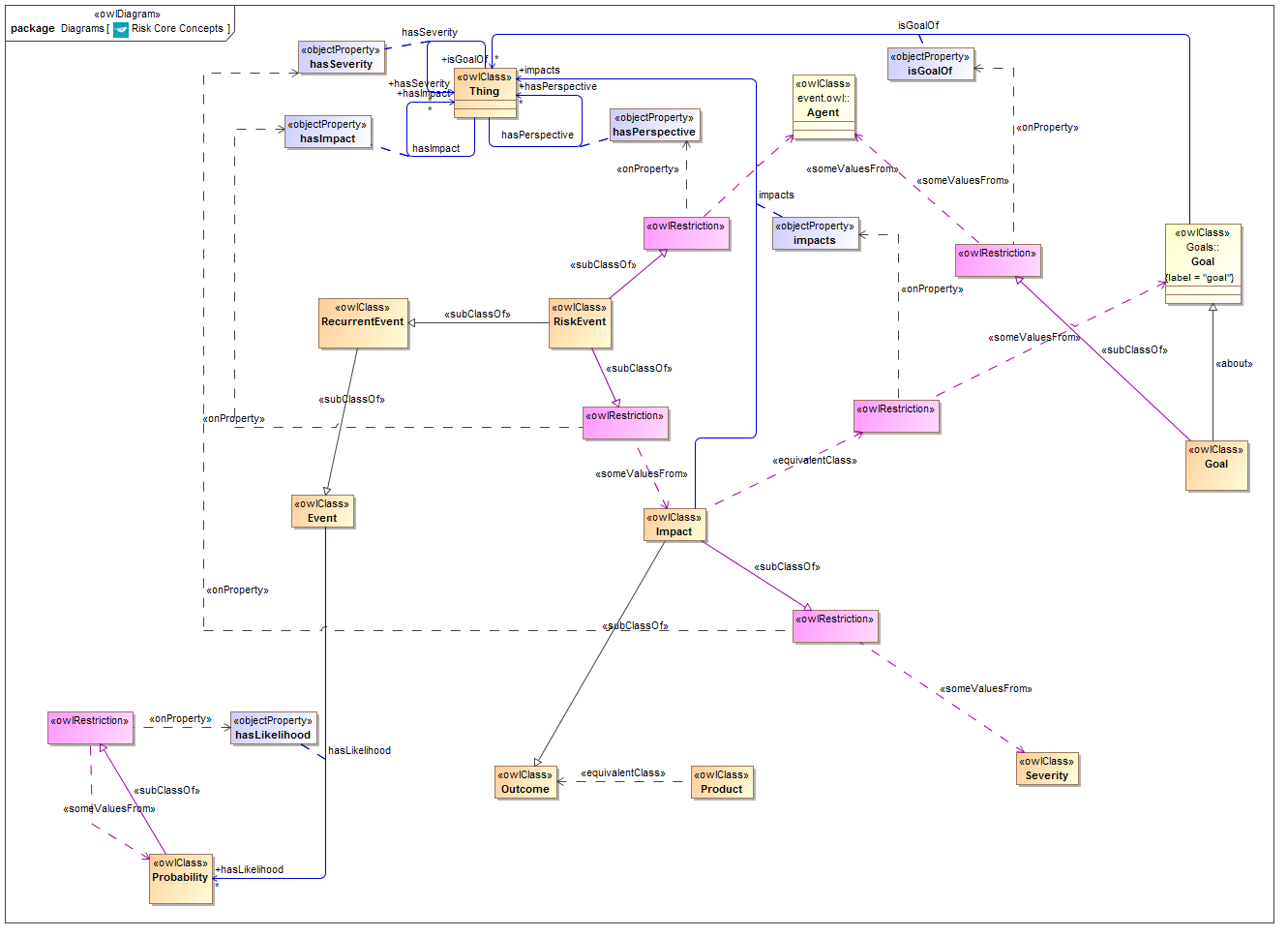
# Extended Content Static View

This means the inclusion of OWL Restrictions in a business-facing way. See also Ref 1 for additional exploration of this including ideas for the rendition of complex combinations of restrictions.

The requirements shown here for the “Static View” are also those renditions which we hope to be able to achieve (within reasonable limitations) in the “interactive” drag and drop functionality for creating these.

That is, there are cases where a restriction or sequence of restrictions already exists and we want to have this inspected and automatically rendered as described in this section; and then in the later section we consider what it will take to create these afresh in front of a live SME business audience.

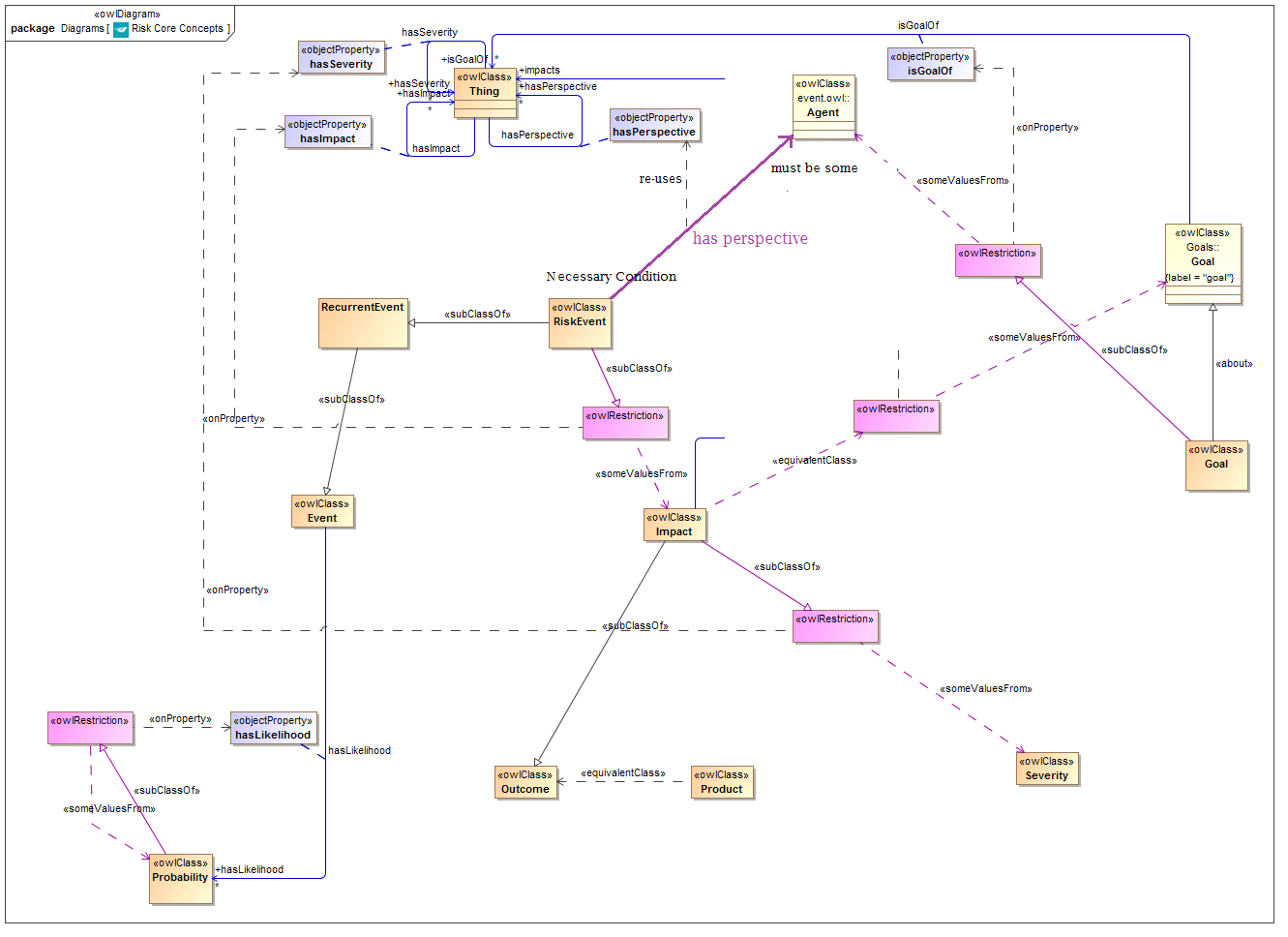
Figure 2 shows a diagram with OWL Restrictions rendered using full ODM constructs:



**Figure 1: Existing ODM rendition of OWL Restrictions**

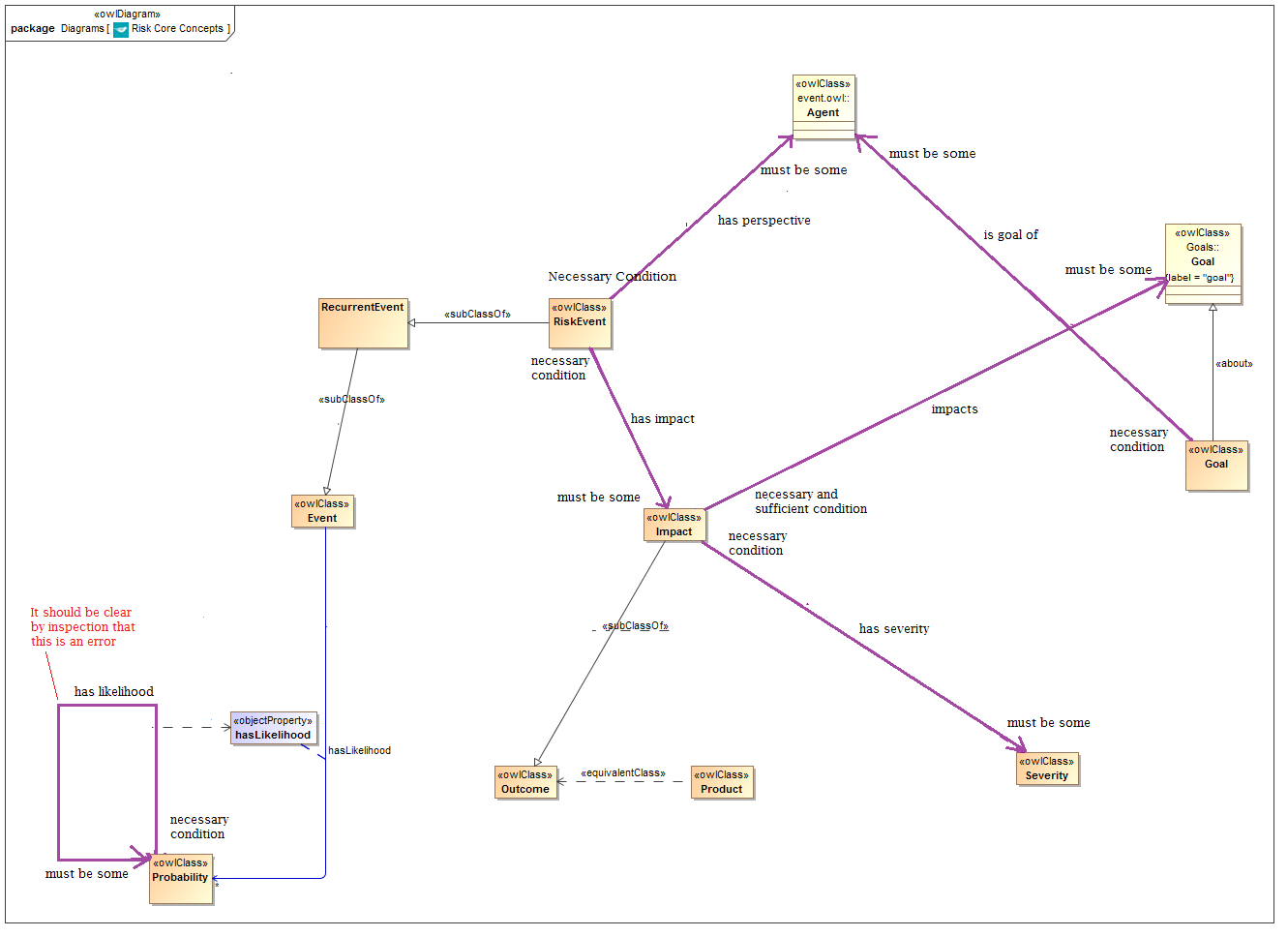
An example of a restriction is circled. The diagram contains several others. Unlike the OMG specifications, here we have already gone halfway to the new requirement by rendering these in a unique color and lining up the separate elements of the pattern. The OMG diagrams don’t do this.

Figure 2 shows a possible pattern for replacing these with a business-facing presentation.



**Figure 2: Possible alternative Restriction rendition**

* It must also be possible to show this WITHOUT the original property being on the diagram. Then the user just sees a line with the name of the property which they selected from the drag and drop toolbar (in interactive mode), and it looks just like any other property except for its color and the additional labels.
* Figure 3 shows what this might look like…



**Figure 3: Restrictions rendition without the original properties**

Note that an error in the original ontology (which was imported into VOM from OWL) is apparent to the viewer as a result of using this notation. In this instance the original property is shown because it is properly a part of the current ontology.

### Aliasing Label Values

Specific features of the proposed business-facing pattern correspond to specific features in the OWL syntax, which correspond in turn to specific UML constructs with their stereotypes, as per ODM.

Figure 4 shows a summary of the relationships between these for a specific type of restriction. The similar equivalences can be found in Tables 1 and 2.

Table 1 and 2 show what text the end user sees, versus what is the corresponding type of the corresponding ODM UML element (dependency etc.).

Thing

Range of

onProperty

involves

refinement of

Cardinality OR restriction type

(re-worded)

Relation to Class

(reworded)

Name of range of

onProperty

at least 1

**Controlling Party**

necessary

condition

Control

involves

Range of onClass, valuesFrom, someValuesFrom or allValuesFrom

Range of Relation to Class

**Figure 4: Derivation of the components the business facing notation**

#### For the “sharp end”[[1]](#footnote-1):

**Table 1: Types of ODM UML Dependency stereotypes and their replacement text**

|  |  |
| --- | --- |
| **Restriction Type**  Stereotype of UML Dependency which has the restriction class at the blunt end | **Business Meaning**  Relationship end label on the sharp end of the arc |
| <<someValuesFrom>> | must be some |
| <<allValuesFrom>> | may only be |
| <<onClass>>  (or <<valuesFrom>> if seen) | *This is a cardinality restriction: additional details from the below are to appear on the sharp end* |
| Qualified Cardinality | exactly / at least / at most / between n..m (of the class shown) |
| No <<\*valuesFrom>> or <<onClass>> relation | *This is an unqualified cardinality restriction – see below* |
| Unqualified Cardinality | exactly / at least / at most / between n..m (of the range of the property itself) |
| minCardinality = 0 | may be some (of the range class) |

#### For the “blunt end”

**Table 2: Types of ODM UML relation to the class, with their replacement text**

|  |  |
| --- | --- |
| **Restriction Application**  Type of UML relation between the Restriction class and the class which is at the blunt end of the new visual construct | **Business Meaning**  Relationship end label on the blunt end of the arc |
| UML Generalization: <<subClassOf>> | necessary condition |
| UML Dependency: <<equivalentClass>> | necessary and sufficient condition |
| Neither (no such relation)  (restriction is free-standing and is the target of a <<\*valuesFrom>> or <<onClass>> relation) | No text.  *Note that this feature requires further exploration – see Ref 1 for tentative rendition. Would almost certainly require the existence of some kind of node e.g. a class box.* |

#### UML Note:

* There may be one, two or three UML Dependencies having the Restriction at their blunt end:
  + <<onProperty>> (always present)
  + <<someValuesFrom>> or <<allValuesFrom>> or <<onClass>> (0 or 1 of these)
  + <<equivalentClass>> if present then there is no UML Generalization <<subClassOf>>

Hence it is not always possible to use the presence or absence of UML Generalization alone to determine types or graphical treatment. Stereotypes must be detected.

### Cascaded Restrictions: Discussion Point

The types of restrictions indicated in row 3 of Table 3 (those without either a generalization or a dependency relation) present a special problem. These indicate the presence of a cascade or restrictions, where the sharp end of one restriction (the <<\*valuesFrom>> or <<onClass>> relation references another restriction. There are several patterns of such restrictions in the submitted FIBO OMG specifications.

Reference 1 includes a detailed exploration of ways to present these. Figure 5 give a simple example. However this presentation remains unsatisfactory for a business audience. Also the cascade of restrictions in question did not actually work from a classification perspective (it caused classes to be misclassified by the reasoner).

#### Discussion

We should therefore consider alternatives, such as limiting the kinds of restriction cascades which are used, to a sub-set which not only can be meaningfully presented to SMEs, but which have been tested and shown to not have problems. It may be that as new cascading structures are developed and tested during the FIBO OMG architectural transformation process, that any patterns which are shown to work are then subject to some development of graphical patterns for presentation to SMEs, and not before. Alternatively, it might be reasonable to simply ban these altogether and ensure that the problematic node in the middle of a cascade such as the one in Figure 5, must itself be a meaningful class of “Thing” with a name and a business definition.

**Thing**

at least 1

controls

*that which*

must be some

has role

necessary

condition

**Controlling Party**

**Figure 5: Cascading Restrictions example**

### More Advanced Renditions

There are two patterns used extensively in the original EA models which should map to a given combination of Restriction and Logical Union. These are:

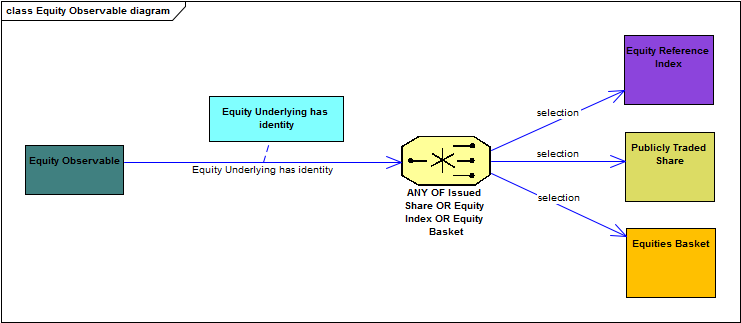
* A property line with an “Any Of” logical combination of target values (ranges)
* A property line with a “One Of” logical combination of target values (ranges)

These are rendered using something like an XML Spy logic glyph, though alternative renditions are acceptable (e.g. the BSI Logic boxes, though these are less self-explanatory than earlier BSI versions); of a box with an Ampersand or the word OR and so on.

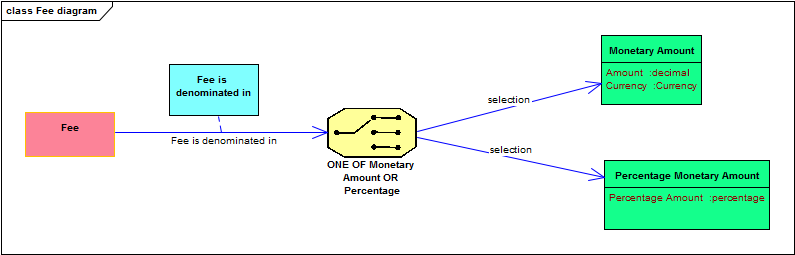
#### Some examples from the EA model.

* What they should look like in Cameo Business Facing View: similar to the EA diagrams
* What they should become in the underlying OWL model
  + Cardinality Restriction with cardinality=1, with <<onClass>> or <<valuesFrom>> relation having a range of a logical union of those classes
  + Restriction with <<someValuesFrom>> relation having a range of a logical union of those classes
* In either case, a Logical Union is created in the underlying VOM constructs, and the lines drawn (dragged and dropped, in the interactive rendition) from the logical glyph, become “unionOf” stereotypes on UML Generalizations, drawn FROM the classes that the lines were drawn TO, and TO the Logical Union.

Figures 6 and 7 show the existing business-facing renditions of certain logical combinations of possible ranges for a property. These need to be replaced with patterns of restrictions and logical unions (the current OWL simply replaces the logical glyphs with OWL Unions, which are necessarily anonymous).

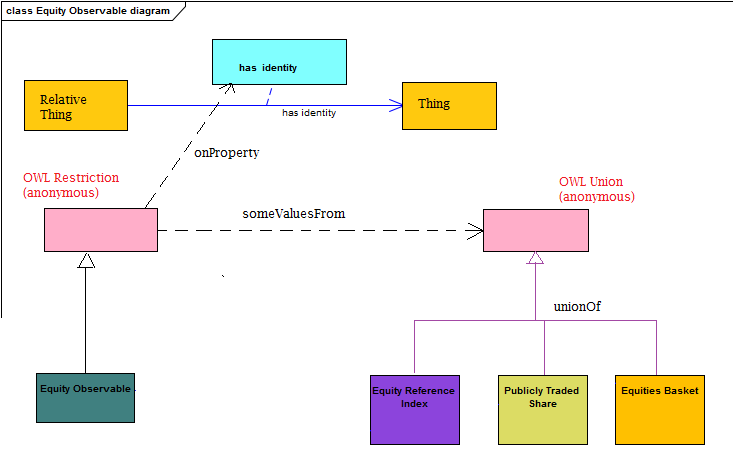


**Figure 6: Example of an “Any of” logical combination as presented to SMEs**



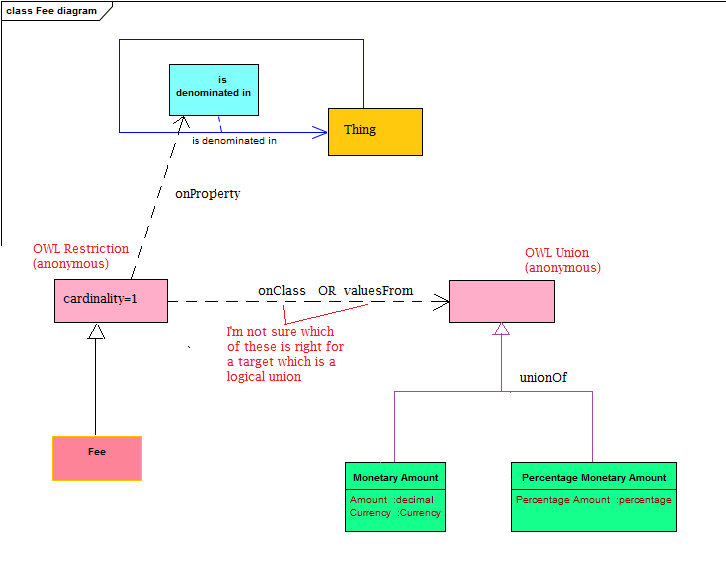
**Figure 7: Example of a “One of” logical combination as presented to SMEs**

Figures 8 and 9 show what would be the appropriate rendition of these in the OWL language, as represented in ODM (we could provide RDF/XML text representations for these examples if that would help).



**Figure 8: Example ”Any of” combination in Figure 6 as OWL Restriction**

(ignore the colors)



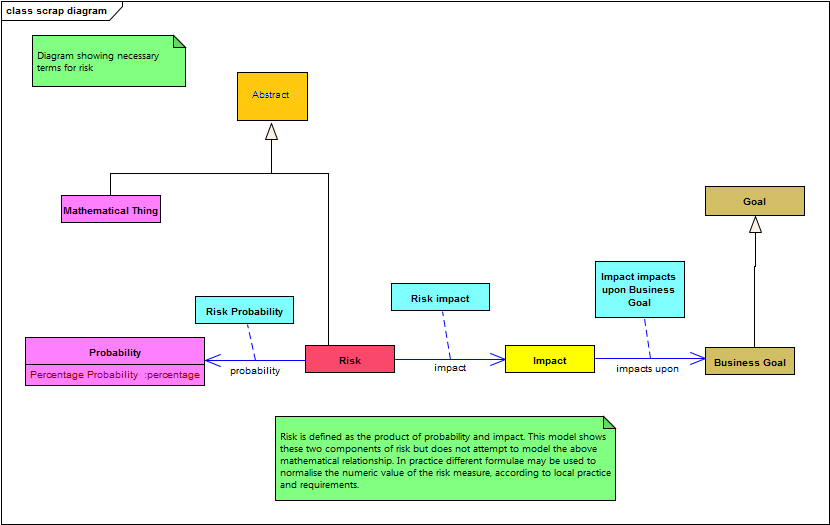
**Figure 9: Example ”One of” combination in Figure 7 as OWL Restriction**

# Basic Content Interactive Functionality

This is basically a demo of what we can do in EA. The functions are:

* Drag and Drop of OWL Classes
* Drag and Drop of OWL Properties (Object Property and Datatype Property)
* Drag and drop of sub property and sub class relations
* Creation of a set of generalizations (UML GenSet) with the stereotype of <<unionOf>> (need to check the name there)
  + The EA feature here could be improved upon.

Figure 7 shows an example of a simple diagram created in EA. This is an archetypical diagram (ontology pattern) and these are typically extended into arbirtrarily complex diagrams during SME reviews, before being simplified again (by drag and drop from the EA equivalent of the Containment Tree) for onward publication and further SME reviews.



**Figure 7: Example EA diagram**

### Details

* Classes and Properties
  + Drag and drop of existing classes and properties from Containment Tree
  + Creation of new classes and properties from the / an OWL-ODM toolbar
* Disjoints, inverses
  + Automatic rendition of any existing disjoints and inverses when elements that have this relationship between them are placed on the diagram canvas
  + Creation of disjoints and inverses from toolbar
* Metadata Editing
  + Ability to view and edit definitions
  + Ability to view and edit <<editorialNote>>, <<explanatoryNote>> etc.
  + Ability to view synonyms and generate additional synonyms
  + Ability to review and edit provenance metadata

In general, it should be possible to access, view and edit these metadata elements with a minimum of opening of technical-looking dialog boxes (though this can be tolerated).

Ideally, metadata which already exists should be displayed in some kind of text window. As a minimum the definition element should be so displayed. Being able to enter editorial note and explanatory note in an open text edit window e.g. near the diagram, would be ideal. In EA these are all in a UML Notes element tied to the element (class or property) and displayed in a window.

Displaying the definition on a mouse hover (per EA versions 9 and above) is also a huge benefit (SMEs have seen it and like it).

# Extended Content Interactive Functionality

* Visual aliasing as for Static View
* Drag and Drop of the Restriction element

## Ideal behavior:

* Toolbar contains an element which is the Property on which the restriction is to be placed
* Drag the property into the diagram
* Drag from “domain” to “range” (these are not really OWL domain and range but that’s how the user sees them)
* Restriction alias (which looks like a property) is created with the blunt end planted in the “domain” class and the sharp end in the “range” class or datatype element
* Name is the name of the property of which this is <<onProperty>> and is displayed as such next to the association edge
* Blunt end: selection option for “necessary” or “necessary and Sufficient”;
  + Selected value becomes a label on this end
* Sharp end: selection option with three options:
  + may only be
  + must be some
  + multiples of
    - if “multiples of” selected, additional selection or entry of the cardinality (without using that long word!)
    - This is to be offered in plain English.
    - Note that as well as being able to select “one or more”, “zero or more” and so on, user must be able to type in a number e.g. 2 or 12, against text saying “at least” and likewise “at most” and in ranges when they want to specify Max and Min values together.
    - Probably best to give them a box for Minimum and another for Maximum and let them either select 0, 1 or “any” OR insert their own value (i.e. “other”).
  + Ideally these are an interactive drop down graphic element. Use of a common dialog box would be acceptable at least in the short term.

### Under the Hood

* Restriction created with “onProperty” relation referencing the property which has the same name as the selected toolbar element
* When planting the “blunt end”, relationship from that restriction created, which is either
  + a generalization stereotyped <<subClassOf>> FROM the selected class, TO the Restriction (necessary condition)
  + a dependency stereotyped <<equivalentClass>> FROM the selected class, TO the Restriction (necessary and sufficient condition)
  + If it’s easier, make the user select between “necessary” and “necessary and sufficient” before this element is created under the hood.
* When planting the “sharp end”, relationship from that Restriction created, having target of the selected class or datatype, which is one of the following corresponding to the selection:
  + “may only be”: <<allValuesFrom>>
  + “must be some”: <<someValuesFrom>>
  + “multiples of”: <<onClass>>

and the Restriction becomes a Cardinality Restriction

* When selecting the multiples in the Cardinality Restriction, the selected (or written) result becomes the Cardinality. This is to be offered in plain English, while the under-the-hood functionality sets this according to the rules for OWL cardinalities.

# Archetypes

This is a further extension of the capabilities above. It requires “double stereotype” recognition.

Details to follow in a future version.

These may be inspected in the EA model. The colors on the diagrams are derived from these.

Metadata to support this in VOM is still under discussion

End result is a set of toolbars with actual business concepts, which the user can drag and drop, creating OWL classes and properties with the archetype stereotypes.

Note that archetype properties may not be appropriate if we have drag and drop placement of property restrictions – the properties being restricted would effectively correspond to the archetypes.

1. The terms “domain” and “range” as used in OWL are not correct when referring to the ends of the business facing element line proposed for restrictions. Therefore a clearly non-technical terminology scheme is used in this document: “blunt end” and “sharp end”, being the elements which are to be regarded as the domain and range of the restricted property, and which are respectively non-arrowed and arrowed. These terms are also used for supplier and client ends of UML relationships. [↑](#footnote-ref-1)