Workstream 2: Technical / Modeling Framework

**Session Notes**

**Thursday 12 July 2012**

# Overview

Workstream 2: Technical and Modeling Issues workstream this week addressed the question of short v long object property names.

The EA-based conceptual ontology uses long and unique names for each object property, and uses each object property only once (for display reasons which are explained in this note). When transforming to operational ontologies it would be desirable to have shorter names, and re-use object properties wherever the same meaning is asserted. It is also intended that we rationalize the object properties themselves so as to have simpler properties used more widely.

There is a question as to whether to deal with this as part of transforming ODM-based conceptual ontologies into RDF/OWL operational ontologies, or whether there is a way to address this question in the conceptual ontology itself. In the latter case, this would need to be done without compromising the requirement to be able to generate business-facing diagrams of the conceptual model content. There is currently no mechanism for presenting the concept of “restriction on domain” or “restriction on range” in a business facing way.

As it turns out, it may be possible to meet this requirement in Adaptive, i.e. presenting each set of restrictions on an object property, as a unique line between classes.

These notes summarize the issue as we find it, and decisions made on the above.

## References

Finance Domain Task Force June 2012

|  |  |  |
| --- | --- | --- |
| **Doc Number** | **Title** | **Filename** |
| Finance/12-06-01 | FIBO Foundation (convenience doc) | Finance-12-06-01.doc |
| Finance/12-06-02 | FIBO Business Entities (convenience doc) | Finance-12-06-02.doc |
| Finance/12-06-03 | Value Delivery Modeling Language (VDML) Overview for the Finance Task Force | 12-06-20 VDML for Finance - 2.pptx |
| Finance/12-06-06 | FIBO Usage Scenarios | FIBO Usage Scenarios.pptx |
| Finance/12-06-07 | FIBO Release Planning | FIBO Release Planning.pptx |

Contents

[Overview 1](#_Toc332283565)

[References 1](#_Toc332283566)

[Headline Points 3](#_Toc332283567)

[Introduction 4](#_Toc332283568)

[Background 4](#_Toc332283569)

[Rationale 4](#_Toc332283570)

[Object Property Rendition in RDF/OWL 5](#_Toc332283571)

[Discussion 5](#_Toc332283572)

[Discussed Option 3 – making predicate names unique. 5](#_Toc332283573)

[Debt Securities Object Properties 6](#_Toc332283574)

[Debt Principal Repayment Terms 6](#_Toc332283575)

[Bond Interest Terms 7](#_Toc332283576)

[General Observations 8](#_Toc332283577)

[Sub Properties 8](#_Toc332283578)

[Findings 8](#_Toc332283579)

[Proposals 8](#_Toc332283580)

[Types of Sub Property 9](#_Toc332283581)

[Proposal (Adaptive) 9](#_Toc332283582)

[Applying to Business Entities Ontologies 10](#_Toc332283583)

[Actions Summary 10](#_Toc332283584)

[Naming 10](#_Toc332283585)

[Rendition of Restrictions in Adaptive 11](#_Toc332283586)

[Summary 11](#_Toc332283587)

## Headline Points

* Looked at the more extensive but older material in the Debt section of the model to explore these issues
* Also looked at the Business Entities section
* Will try to use unique predicate names for object properties in the Business Entity section
  + MB to identify possible unique names and revert to the group
  + We will then see if there are cases where we need to consider e.g. predicate + Range as a unique naming mechanism
* Adaptive has a means for showing re-uses of an object property as separate simple lines between classes
  + To be tried out on a defined sub set of the terms we have in the model
* At the same time, we intend to rationalize the top end of the object properties in any case, and align these with concepts from GIST, DOLCE and elsewhere
  + (this has been discussed for some time – hope to look at this over the summer)
* Model can then distinguish between
  + properties which introduce a distinct meaning (refinement or restriction of existing ones) and
  + properties which are not unique or which differ only in the localization of the definition and mapping to conventional data model properties (per ISO 20022, in Securities models)
* This would be a considerable enhancement to the model as it stands.
* This will be worked on over the summer with a view to having some firm proposals for the next draft “convenience document” for business entities in the September time frame.

# Introduction

This session was dedicated to the question of object property (relationship fact) names.

## Background

Currently the FIBO model in EA has two styles of names for object properties:

* A complete, long name usually (but not always) in the style of ‘Domain predicate Range’
* A label on the relationship line which consists only of the predicate.

There are two issues with this:

1. The ability to present diagrams with just the short predicate names (as done for the business-facing “simple” diagrams) relies on a bug or UML non-conformance in EA whereby Association Classes may have a separate name for the class component and the association component;
2. The resultant names are sometimes very long indeed, and in some OWL or SBVR applications these apparently get concatenated (again) with the domain and range names.

Several options have been explored:

* Shorten the name to
  + Domain + predicate; or
  + predicate + Range
* Make all predicate names unique

At present, in the material for FIBO-BE plus the half of Global Terms that are currently included in FIBO-Foundations there are approximately 70 non unique predicate names – a manageable amount? There will of course be others in the more extensive material for securities, derivatives, market data etc.

## Rationale

One difference between the modeling approach taken in FIBO and that usually applied in RDF/OWL models, is that we have used each object property only once, with one domain and one range.

This is a requirement in order that diagrams, which are intended for business consumption, may have a single line between the box which represents one class (the domain) and the box which represents another class (the range).

Anything other than this is not considered appropriate to present model content for business domain consumption. For example, it is possible to explain to people that an object property represents a relationship and that this relationship may apply between more than one pair of classes, however to explain this you are bringing the viewer into the world of the modeler, and in so doing, you are moving away from the world of self-explanatory “boxes and lines”.

It is hoped that at some point in the future, it would be possible to find ways of presenting this kind of arrangement in a way which is immediately self-explanatory and requires no grounding in or explanation of model theory or semantic modeling. For this reason, we are not wedded to the idea of having each object property expressed once and having a long unique name – this is simply a feature of the business layer presentation requirements as we see them.

## Object Property Rendition in RDF/OWL

In rendering this material into RDF/OWL there are two points at which we need to think about naming:

1. In the RDF/OWL rendition of the complete conceptual model content;
2. In individual RDF/OWL operational ontologies that are built for specific use cases

It would be desirable to come up with a naming solution that allows for each object property to be used once and have a single name, without the long name complications of the current naming system; this would be reflected both in the ODM repository and in (1) above. Then there would be no requirement to make any changes when deriving operational ontologies (2) from these. That is the preferred outcome of this investigation.

Whether or not this is possible, we want to be able to transform this model content (eventually in an automated way), to RDF/OWL operational ontologies on a per use case basis, with re-used object properties in the design pattern usually used in RDF/OWL ontologies. We would therefore need to figure out how to identify the reusable object property, and for example see if the archetype of each object property in the conceptual ontology can become simply the object property in the operational ontologies.

In parallel with this investigation, we want to revisit the object property archetypes and align these with the best of what we can find in existing upper ontologies, in particular GIST and DOLCE. At present the naming and the archetypes of these object properties has been somewhat reactive.

# Discussion

Several options exist:

1. Shorten the name to Domain + predicate;
2. Shorten the name to predicate + Range
3. Make all predicate names unique and use only the predicate

## Discussed Option 3 – making predicate names unique.

Discussed this option. Consensus was that if this is achievable this would our preferred option. If it is not achievable we would have to consider (1) or (2) above.

There are 70 non unique names just in the FIBO-BE material, which is much smaller than the securities or derivatives models.

Looked at the securities models – focusing on the Debt section as this is the most extensive.

## Debt Securities Object Properties

Looked at relationships to interest payment terms, and to redemption / call terms, from Debt Instrument and from “Bond”, which is a sub-class of Debt Instrument.

### Debt Principal Repayment Terms

We have:

Between the class “Bond” and the class “Bond Principal Repayment Terms Set”:

* Object property: **“Bond has Principal Repayment Terms”**

**Definition: “**Repayment terms for the Principal on a Bond”

Note that the label here is “has principal repayment terms” rather than simply “has” as per the long name. This was done before we standardized the approach described above and used in Business Entities. As a result it has a unique predicate name which is a lowercase copy of the predicate “has” and part of the range name. In later parts of the model, this would have been styled “has terms”, which is a relationship label used extensively in the securities and derivatives models.

This has several sub properties, one of which is:

* Object Property: **“Amortizing Security has Amortizing Terms”**

**Definition:** “Terms for the Amortization of the Principal on an Amortizing Security; the amortizing security terms and conditions.”

This has the label “has principal amortization terms”

This has two further sub properties:

* From the class “Sinking Fund Bond” to the class “Sinking Fund Amortization Terms Set”
  + Labeled “has principal amortization terms” as above
* From the class “Index Amortizing Bond” to the class “Index Linked Principal Determination Terms Set”.
  + Labeled “has principal repayment terms”

Note here that the labels are similar but different, and identify whether the terms are terms about repayment of principal or terms about the amortization. However, this amounts to much the same thing – amortization is a means of repayment. Nearby, there is an Inflation Bond, which is not a direct sub-class of Amortizing, but which has a similar relationship.

This part of the model is complex: during SME reviews it was determined that some kinds of bonds amortize via a “Sinking fund”, while others have or may have amortization terms linked to a variable index (this was difficult to verify – this came from a data model in which index and amortization terms were combined for efficiency there was some uncertainty about which of the logically possible terms actually exist in practice). We later established that for Inflation Bond, it is not the amortization which is linked to some inflation index, but the actual principal amount.

Hence there are a lot of subtle distinctions of similar object property semantics in this part of the model. Where these are truly separate meanings, it has already been shown possible to give these unique predicate names.

#### Observations:

Here we do in fact have unique predicate names: “has principal repayment terms” and “has principal amortization terms”. Other sub-properties of “has principal repayment terms” include “has repayment terms” (this for a bullet bond), which could with a little imagination be made more self-explanatory.

#### Conclusions:

* This can be rationalized – these names do not really follow any consistent naming pattern at present
* It is possible to give these relationships unique names based on the predicate
* These ones do imply unique meanings which are related by sub property relationships

Note also that if these were named formally as predicate+Range, then the name would be more ‘clunky’ since the ranges here are contractual terms, which have been named “Terms Set” in order to comply with the naming convention whereby all OWL class names are in the singular.

***Aside:*** There is a parallel proposal on the table to replace all classes that represent sets of terms, labeled “Terms Set” with classes labeled simply “Term”. This has not been implemented in the Debt or other securities sections we were looking at here, and would be part of the changes to be made globally before those sections are taken forward into the OMG process.

### Bond Interest Terms

Found a hierarchy of terms between Bond and Coupon that are relevant to this investigation:

* + Fixed Coupon Bond has coupon Bond Fixed Coupon
  + Variable Coupon Bond has coupon Bond Variable Coupon
    - Floating Rate Note has coupon FRN Coupon
    - Index Based Coupon Bond has coupon Index Linked Coupon

**Observation:** These all have the same predicate label “has coupon”

**Definitions:** these have the same definition (seemingly derived from ISO 20022) with minor variations.

**Conclusion:** in this instance, all of these relationships represent a single object property, with no discernible variation in meaning and little difference in definition other than to refer to what is unique about the range of each one. This is a good candidate for the treatment whereby we would have a single Object Property and have restrictions on the range.

## General Observations

This is a much older part of the model. The naming convention of ‘Domain predicate Range’ has not been followed in all cases. In many cases, there is a relatively unique predicate name for the relationship.

### Sub Properties

Many of these object properties relate to attributes in the ISO 20022 model. They specialize the meanings of the more general concepts (e.g. ‘Floating Rate Note has coupon FRN Coupon’ specializes ‘Variable Coupon Bond has coupon Bond Variable Coupon’).

In the ISO 20022 model, for the most part these properties are given as optional terms of the more general “Debt Security” and are not specialized in this way. The classification of various instruments in terms of cashflow types such as fixed v variable interest, and bullet v amortizing repayment, is effected by the provision of refined or restricted properties for interest and principal payment terms which define those classifications.

In other cases, these terms do represent individual terms within the ISO 20022 model, with their own definitions.

## Findings

In some cases, the meaning remains the same but the existence of the separate object property means that the definition which applies for the data model term in e.g. ISO 20022, is the definition unique to this relationship. That is, the use of the non-OWL pattern of single occurrences of each relationship, makes for a good one to one mapping between the semantic model and the corresponding data model.

# Proposals

It should be possible to identify when these relationships introduce a genuine new meaning as a specialization of the existing meaningful term as it applies to the parent class, and when they do not introduce new meanings, and are only present so that a localized definition may be applied.

Looked at the definitions of these properties. Some of these are really just copies of the definition of the parent property, with the localization of the range given in the definition e.g. “has principal amortization terms” for a Sinking Fund Bond simply enhances the definition of “has principal amortization terms” for an Amortizing Security, with little change in wording. This implies that, in these cases at least the meanings are not really unique, and in the OWL world it would be possible to frame this relationship as a restriction on the range of the parent relationship.

In other cases, the definitions are clearly drawn from the written definitions of the data model terms (e.g. ISO 20022 attributes or associations), and these imply at least some specialization in meaning. That is, in these cases we have taken a term in some source data model, and identified what relationship fact (object property) it can be framed as a specialization of.

**Action:** Review these relationships and come up with common patterns. We expect to find that for any given hierarchy of relationships, one or another of the following will apply:

1. The sub property introduces no new meaning, and the definition is localized in name only, i.e. it is more or less a copy of the definition of the parent with some acknowledgement of the reduced range
2. The sub property is a distinct, meaningful term, which has been determined in the conceptual modeling as being a sub property of some other, meaningful but broader term
3. Things are not so clear cut: the term has a unique meaning and a correspondingly unique definition drawn from some data model standard, but it would be acceptable to regard it as being a separate application of some more general term such as “has part” or “has terms”.

## Types of Sub Property

**Proposal:** Can we identify which object properties introduce some new meaning and which ones do not, and identify the appropriate treatment for each?

On the above object properties, sometimes these were the same meaning with some restriction on the domain and range, and in other cases these are distinct meanings. Would it be possible to go through the model and identify which is which?

We could consider some kind of annotation to distinguish these, so that we can still present each relationship as distinct lines on the ODM / UML model diagrams, but be able to identify which ones are to be rendered in OWL as the same object properties with restrictions. Note that restrictions cannot currently be indicated in the UML-based diagrams for business audiences.

**Naming:** for the most part, the existing predicate names should be a clue. Where these are the same for two object properties, it is likely that these should be the same object property with restrictions. Where they are different, it is likely that the relationship introduces some identifiably distinct meaning as indicated by the predicate name.

From this we can not only distinguish which is which, but we can also identify unique predicate names so that the naming convention of predicate-only can be followed.

## Proposal (Adaptive)

**Pete:** It may be possible to render the diagrams in Adaptive in such a way as to show a single line from one class to another, where that line represents a restriction on an existing relationship which is also shown as a single line between the more general classes.

**Discussion:** this is a possible game changer. It means that we would be able to render diagrams in line with our established business presentation requirements (i.e. lines between boxes, with no modeling knowledge required to interpret those lines) while following the convention of reusing simpler properties multiple times.

## Applying to Business Entities Ontologies

Discussed how to apply this to the current material i.e. Business Entities models. Note that these were not derived from ISO 20022 or other data model standards.

**Discussion:** can we find a suitable set of relationships which are in a sub-property hierarchy?

There are two that would be suitable:

* The “constituted by” relationships that relate a kind of formal legal person (Artificial Person) to the relevant kind of incorporation document by which they are constituted (e.g. Memorandum and Articles for Incorporated Companies);
* The “governed by” relationships that relate a “Formal Organization” to the contractual instrument by which it is governed (a contract between the principals, for example a Trust Agreement).

In both cases, the meaning is broadly the same but the definitions have been drafted to reflect the precise relationship, including reference to the domain (what kind of entity is constituted by or governed by the relevant contract or instrument) and to the range (what kind of instrument it is).

**Agreed Actions:**

* Mike will send details of these relationships to Pete Rivett, who will look at how these object properties may be modeled in Adaptive

# Actions Summary

### Naming

**Action:** MB to go through the model and identify relationships which are the same concept with some restrictions to domain and range, and relationships which are distinct concepts.

**Action:** MB to go through the model and allocate unique predicate names to those relationships which are identifiably unique.

Once this is done, we should have a clearer idea of where we stand with regard to reusable object properties in the RDF/OWL notation, and how to identify these in the ODM UML-based model.

We will then be able to identify whether it is possible to give each object property a unique predicate-only name. If not, we may revert to the predicate + range notation for object properties which are not unique and which are restrictions on range only. A similar thing may apply to Domain + predicate, i.e. it may be possible to have a unique naming scheme which identifies what it is that makes that particular object property unique. Where the uniqueness is in the domain or the range only, this will indicate that in the RDF/OWL rendition of the same model content, the object property should be rendered using restrictions to that identified domain or range, as a restriction of the object property identified by that unique predicate name.

### Rendition of Restrictions in Adaptive

* Send the relevant model material to Pete Rivett (MB)
* Put forward proposed graphical rendition of these in Adaptive (PR)

# Summary

We believe it should be possible to come up with unique names for most or all of the object properties we have, based on predicate only.

There may be cases where the property is really a restriction to an existing property; if so, we will need to explore naming further (we didn’t reach a consensus on how to deal with these), but it’s likely these can be rendered as ‘predicate + Range’ or ‘Domain + predicate’ depending on which it is that is restricted (and maybe if both are restricted, we use ‘predicate + Range’ for brevity). The precise approach will depend on the outcome of the first pass at rationalizing these names. The results of this will be presented at our next session (not in August as we are on recess this month), and will be circulated by email once this exercise is done.

Meanwhile, it should be possible to render relationships in Adaptive as single lines, which are restrictions on domain and range of other relationships. This would significantly simplify the overall issue, though we would still need to consider how users of the ODM based modeling tools (including EA) would be able to model in this environment and draft proposed extensions to the model, while rendering these in a business-consumable format.