Workstream 4 October - Conformance

# Summary

This note summarizes the discussions on the “Conformance” section of the FIBO Foundations and FIBO-BE formal specifications. These sections will set out what applications may be asserted as being “Conformant” to the FIBO specification, and what aspects or parts of these specifications these may assert conformance to. This is one of the most important aspects of any OMG specification, and particular attention has been paid to this during formal reviews by AB members and others at the OMG. Previous comments on these sections were to the effect that this material was too complex and far reaching. Discussions at the June 2012 OMG FDTF meetings gave us some simpler ideas to try, and today’s discussion is intended to finalize what will and what will not go into the Conformance sections of both FIBO Foundations and FIBO-BE.

Contents

[Summary 1](#_Toc337719926)

[Background 3](#_Toc337719927)

[Discussion 3](#_Toc337719928)

[Ontology Only Conformance 3](#_Toc337719929)

[Conformance choices: 3](#_Toc337719930)

[Consensus: 3](#_Toc337719931)

[Logical Data Models 4](#_Toc337719932)

[Previous experience in conformance for content-related standards: 4](#_Toc337719933)

[Further Discussion 4](#_Toc337719934)

[Conclusions 5](#_Toc337719935)

[This Appendix 6](#_Toc337719936)

[Overall: 6](#_Toc337719937)

[Conformance Matrix 6](#_Toc337719938)

[Principles 6](#_Toc337719939)

[Model Semantics Conformance 6](#_Toc337719940)

[Semantic Conformance 7](#_Toc337719941)

[Principles 7](#_Toc337719942)

[Details 8](#_Toc337719943)

[Notes to the Table 10](#_Toc337719944)

[Examples 10](#_Toc337719945)

[On Disjoints and Taxonomies (1 and 2) 10](#_Toc337719946)

[On Properties 11](#_Toc337719947)

# Background

We have considered two possible aspects of the Conformance material for the specification:

* Minimal conformance points based on the use of individual ontologies in whole or in part, without changes;
* Additional conformance points to cover adaptations of the model content for functional applications

These considerations would be applied, as separate conformance points, both to the use of operational ontologies (in OWL) and of conventional data model designs (OO classes or similar).

We had previously set out a number of basic “Principles” for the latter. It remains an open question as to whether these principles can be defined for the specific technologies in which applications may be written, such that this application of these principles can be set out as formal conformance points in the OMG FIBO specifications. That is, these need to be set out in such a way that non-conformance can be unambiguously defined and identified.

# Discussion

## Ontology Only Conformance

Ideal: just extract individual ontologies from FIBO

Alternative: as per the current Operational Ontology work – where additional parent classes are removed. This raises challenges e.g. you need to reinstate the inherited properties from those classes.

### Conformance choices:

1. Assert conformance only when using individual ontologies (or parts thereof) in the operational ontology
2. Assert conformance for changes in the operational ontology, such as (in this example) collapsing of hierarchies or bypassing to simply “Cheese is a thing” approach.

### Consensus:

Not leave it open to people to collapse things, they either use one or more modules from the BCO or not. That is, you can’t chop out inheritance and still assert conformance.

**Pros:** simplicity

**Cons:** the BCO is not like a normal operational ontology.

**Possible approach:** keep it simple this time, then monitor what semantic tech applications do with it, and identify what if anything are possible conformance points to include in a future iteration of the specification. This could be dealt with in the FTF.

**Consensus:** Agree.

### Logical Data Models

Make these points guidelines for how to develop a LDM rather than making them formal conformance points.

As with the operational ontologies, we can reconsider some of these as possible conformance points during the FTF process.

**Consensus:** Agree.

**MB note:** there are some interesting guidelines emerging for good LDM architectures.

We can’t turn the spec into a tutorial on valid transformations for different modeling techniques. The more we can express the conformance in terms of a set of conformance principles, and then refer to these. Principle-based approach to conformance.

The guidelines illustrate how those principles can be applied in practice for different technologies.

Conformance to principles – desirable if we can do it.

### Previous experience in conformance for content-related standards:

P&C: identifies certain key entities in their model and said that anything conformant must include those key entities.

**Proposal:** Put the principles as well, into the informative appendix of modeling guidelines.

**Consensus:** OK

Then: what does conformance consist of?

There is then no conformance point for a conventional (LDM) application. Effectively, no application would be able to assert conformance, or in other words any application can be conformant (but there is no conformance point they can assert).

**Outcome:** communicate this and give people an opportunity to identify whether this would be a problem. Alternatively, deal with other opportunities for conformant applications in the FTF.

### Further Discussion

However: it should be possible for someone to build a data model that is conformant, and also that someone could try and build a data model that is conformant with FIBO, and assert that it is conformant but it is not.

If the informative appendix is found to have value, then we can elevate it to a new conformance point.

Can such elevation be carried out in FTF? Yes. But not in Revision Task Force (RTF).

Generally: more acceptable to add a new conformance point than to change an existing one.

## Conclusions

**Action:** inform everyone else on this conclusion and get feedback.

**General principle:** If people extend the ontology first, and it still doesn’t blow up a reasoner, then this can be used as the basis for the design of an LDM.

Appendix: Conformance Workings

# This Appendix

This was the “Conformance Clean Sheet” document which we have been working from. This includes articulation of the general “Conformance Principles” included in the earlier formal draft of the FIBO Foundations and FIBO-BE specifications, along with some detailed workings by Mike Bennett on how to articulate the details of these for operational ontologies and for logical data models. From more recent discussions (including today’s 4 October session), it’s been decided that for operational ontologies we want to define conformance simply as the use, as modeled, of any sub-set of the FIBO ontologies as they stand, so one column of these detailed workings would no longer apply.

There are some changes, corrections and notes in this Appendix from the 4 October review session.

# Overall:

* Conformance in Foundations – what makes a conformant application in general
* Conformance in Business Entity – what makes conformant application of this specification

### Conformance Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| With  Of | Model semantics | Model presentation | Whole model content | Individual ontology | ? |
| BCO extension | Y | Y | Y |  |  |
| Operational ontology | Y | N | N | Y |  |
| Logical data model | N | N | N | N |  |
| ? |  |  |  |  |  |

**Comment 4 Oct:** Need definitions for each of the column and row headings.

# Principles

## Model Semantics Conformance

**Objective:** Not have models which make assertions that contradict the semantics of FIBO, while asserting FIBO conformance.

**Scenario:** design a logical data model using the FIBO content as the conceptual model i.e. point of reference for the business semantics required.

there is no reason for someone not to make a logical data model using the FIBO content as the conceptual model without adhering to the model semantics, but if so, this model should not assert FIBO conformance.

In order not to contradict the semantics of FIBO BCO, the following are minimal requirements:

### Semantic Conformance

Points of semantic conformance include:

* Taxonomy and Classification Scheme
  + Organization of classes
  + Adequate abstraction of concepts in to their most general forms
* Correct use of the top level model partitioning

These do not need to be reflected in operational ontologies or logical models, but shall not be directly contradicted by model constructs which have the same effect as the above and which do so with a different impact on the model semantics.

## Principles

These are the basic principles in order to avoid making assertions which contradict those assertions already made in FIBO:

1. It is not necessary to include all the ancestor classes but disjoints asserted between those ancestor classes must be respected
2. Two classes cannot be introduced into the same logical class hierarchy which have ancestors which are disjoint in FIBO. This is because otherwise it becomes possible to introduce contradictions or data structures which correspond to contradictory or untrue (or absurd) facts about the world.
3. Relationships which have restrictions defined for them (for example functional object properties) may not be extended to have looser multiplicity in logical data models but they may be further restricted.
4. New facts or relationships should not be introduced which directly contradict some fact in the FIBO terms which are used, or in any FIBO terms which are not directly used but which *have a bearing* on the terms which are used.

**Comment 4 Oct:** “Have a bearing” can’t be easily defined. Lose (4).

# Details

| **Point** | **Requirement** | **Ontology** | **Conventional (LDM)** |
| --- | --- | --- | --- |
| 1 | No disjoint contradictions | Disjoints may be asserted between classes or descendants of classes that are not disjoint in the BCO  For classes which are disjoint in the BCO, these or their descendants may not be combined into a common class for simplicity | The BCO classes are set theoretic constructs. No class (UML) or equivalent may be asserted as being derived from one such class, which may have instances that are members of another class or classes that is/are disjoint with it in the BCO.  To avoid doubt, classes which do combine instances from more than one class in the BCO where those classes are disjoint, should not have the same name as either or any of the disjoint classes. |
| 2 | No contradictory sub classes | No class may belong to two hierarchies in which some ancestor in one hierarchy is disjoint with some ancestor in the other hierarchy | In logical models, multiple inheritance is not commonly used. A single-inheritance class model is conformant with this requirement by definition.  Where multiple inheritance is used in a data model, no class may belong to two hierarchies in which some ancestor in one hierarchy is indicated in the BCO as being disjoint with some ancestor in the other hierarchy. |
| 3 | No looser properties | Object properties which are directly derived from FIBO may not have a broader range.  Relationships may not be added as a sub-property of some relationship that exists in the FIBO BCO, which have a broader range.  Object properties may not be asserted for a class that is derived from the BCO, which have a broader range than an equivalent object property of that class from which it is derived. | The set theoretic logic of the BCO should not be contradicted by the definitions of classes in a UML Class model or equivalent.  Associations, Aggregations and Compositions may not be added to a class that is derived from FIBO, which have a loose cardinality than the corresponding relationship in some parent class in FIBO.  Associations, Aggregations and Compositions which reflect object properties in FIBO, shall not have a lower cardinality than that of the corresponding FIBO object property.  Classes should not be introduced as children or descendants of a class derived from FIBO, in which there may be instances which cannot be member of the FIBO class due to the properties asserted for them. Since OO classes are not set theoretic, it is possible for the properties to be relaxed (the membership of a class is not defined by its properties), but there must be some formal restriction on what sort of instance data may be instances of this class, for example by the use of some OCL statement. |
| 4 | No contradictory properties | No object property may be added which, if taken in conjunction with the properties already asserted, define a class which is self-contradictory (paradox) or of which nothing may be a member (owl:Nothing empty set class)  No such object property may be added to a class derived from the BCO  No such object property may be added to a new class which is a sub class of a class in the BCO such that the property of the new class contradicts the inherited property of the class in FIBO. | No associations, aggregations of compositions may be added to a class which, when taken in conjunction with some properties of that or a parent class in FIBO (including ones that are not incorporated into the logical model), would result in the definition of an OO class which is absurd or empty.  No class should exist in the logical model which, if all the mandatory properties were reflected by instance data, would not be able to exist [note 1, 2].  No such association, aggregation or composition may be added to a class which is directly derived from a class in FIBO  No such association, aggregation of composition may be present in an OO class which is a child or descendant of a class in or derived from FIBO, or a class which exists in the class hierarchy of FIBO as an ancestor of the class, even if the class in question is not itself included in the logical model.  OO Classes in a logical data model may be formed from a combination of classes in the FIBO BCO, provided only that all the properties derived from the FIBO properties (object property and datatype property) of each disjoint class, have a cardinality of zero or more.  Where a class in the LDM design combines properties from classes which are disjoint in the FIBO BCO, all attributes which are derived from properties in each of the disjoint FIBO classes, shall have a multiplicity of zero or more.  Where a class in the LDM design combines properties from classes which are disjoint in the FIBO BCO, all associations, aggregations and compositions which are derived from properties in each of the disjoint FIBO classes, shall have, on the end distant from the combined class, a multiplicity of zero or more. |

### Notes to the Table

1. Actually there is a case for this, and an important difference between logical models and semantic models: it is common to have one class in an OO model in which there are a lot of optional properties, not all of which would be required to model the data for any one class. For example you may have a class called “Security” which combines facts that are only true of bonds, facts that are only true of shares and so on. This is a valid design decision and often a good one. Should we restrict the design decisions that may be made when crafting a logical design to carry data corresponding to entities defined in FIBO? I would argue not – in which case this conformance point should be relaxed for logical model designs or OO designs generally.
2. A corresponding conformance point which could be asserted, is that there should not be things with a cardinality of 1 or more, that would cause a contradiction of this sort.

## Examples

Here is what we are trying to avoid:

### On Disjoints and Taxonomies (1 and 2)

FIBO has “Independent Thing” and “Relative Thing” as two disjoint partitions.

FIBO asserts that there is a hierarchy of independent things which are Legal Entity, and another hierarchy of “Relative Thing” constructs which are entities that perform some role or are defined by some function. For example, there are entities in FIBO BE which are defined by some function that they perform. These include “Business”, “Non Profit”, “Special Purpose Vehicle”, “Bank”, “Government Agency” and so on. Each of these functionally-defined entities may take the (independent) form of one or more types of entity as defined by their form rather than their function. For example, a business may be a sole trader or a formal organization; a non profit may or may not be a legal person. A government agency may be set up as a legal person or as a less formal body. A Special Purpose Vehiccle must be set up as some kind of legal entity (and probably as some kind of Legal Person), but the law is agnostic as to exactly which form of entity it is.

Having modeled this, we see that Bank, for example, is in the “Relative Thing” partition, and “Limited Company” is in the “Independent Thing” partition.

If someone creates a model in which “Bank” is asserted as being a sub-class of “Legal Entity”, this would contradict the FIBO semantics.

That is not to say this would be wrong. It would conform with a different ontology that has a different semantics for terms like Bank or Legal Entity. However it would not conform with the semantics of FIBO.

Recall that conformance in this document is not trying to define what is right and wrong in an application or an operational ontology; rather, it is defining what is or is not conformant to the semantics of FIBO.

There is also nothing wrong with someone creating an application which uses FIBO as a point of reference for understanding the business domain, but which makes design decisions appropriate to that application which contradict some or another aspect of the FIBO semantics. However, when others go to create applications which must interact with that application, they now know what aspects of the FIBO semantics have been conformed with and what aspects have not.

### On Properties

The aim here is not to introduce some object property which incorrectly changes the semantics already asserted for the corresponding object property of some parent class.

This raises the question of whether properties may be extended (loosened) as well as tightened in child classes.

The principle: classes in child classes may be further restricted (creating a class which is a restriction of the parent class; i.e. membership of the new child class is a proper sub-set of the membership of the parent class).

To introduce relaxations of the properties of a parent class in a child class, would be to make the new child class a set of which individuals could be members, which could not be members of the parent of ancestor classes.

**However:** as long as the model is derived from the FIBO ontology in the correct way, the OWL model constraints should already make sure this sort of thing cannot happen. The conformance question is, where classes in the LDM are formed by combining classes from FIBO, these should not combine elements from classes which are themselves mutually disjoint (like equity and debt) unless all the elements derived from those disjoint classes are made optional. Here we have a design pattern that should probably be allowed.

So:

In an OO model design (LDM), classes may be created which are a combination of two or more disjoint classes in FIBO, provided only that all the properties of all the mutually exclusive classes, are made optional (minimum cardinality of zero)