FIBO Test Strategy

*Version 0.1 initial discussion draft*

*9 September 2014*

**Executive Summary**

**This Document:** Test Strategy outline – provides a jumping off point for development of detailed strategy and test data.

**Stated Requirement:**

* Create full unit test coverage for all the FIBO ontologies (where a unit test is basically a use case represented as one or more SPARQL statements)
* Ensure that these unit tests are executed on as many vendor platforms as possible, enabling vendors to contribute to the further Build of FIBO and allowing them to demonstrate their technology and how FIBO can be applied to the community and the industry as a whole.
* Run use case/SPARQL unit tests and other tests such as OWL validators, reasoned-related tests, performance and volume tests.

This Test Strategy makes reference to a formal set of evaluation criteria defined in a separate document, the FIBO-OOPS!/OQuaRE table [reference 3].

In order to read this document it is advised first to read or skim the “Test Strategy Theory and References” document [reference 2]. That document sets out the thinking behind the things that are described in this document, so as to avoid extra narrative material in this document.

Contents

[Introduction 4](#_Toc398081006)

[Test Strategy Principles 4](#_Toc398081007)

[Applying to Ontology 4](#_Toc398081008)

[The Two Vs 4](#_Toc398081009)

[Validation and Verification Implementation 5](#_Toc398081010)

[Verification 5](#_Toc398081011)

[Validation 5](#_Toc398081012)

[Test Data Considerations 5](#_Toc398081013)

[Verification Tests 5](#_Toc398081014)

[Unit Tests and Regression Tests 6](#_Toc398081015)

[Regression Tests: Unit Test 6](#_Toc398081016)

[Validation Tests and Checks 6](#_Toc398081017)

[Validation Test Data 6](#_Toc398081018)

[Regression Tests: Semantic Enhancements Validation 7](#_Toc398081019)

[Test Strategy 8](#_Toc398081020)

[Use Case of the ontology itself 8](#_Toc398081021)

[Verification Measures (Unit Test) 9](#_Toc398081022)

[TBox Tests 9](#_Toc398081023)

[ABox tests 9](#_Toc398081024)

[Validation Tests (analogous to system and integration tests) 9](#_Toc398081025)

[Semantics and Sensibility 10](#_Toc398081026)

[User Stories and Use Cases 12](#_Toc398081027)

[Population from semi-structured Data 13](#_Toc398081028)

[Refactoring Validation Tests 15](#_Toc398081029)

[Test Namespace Requirements 20](#_Toc398081030)

[Namespaces for the Ontologies 20](#_Toc398081031)

[Namespace for Test Ontologies 21](#_Toc398081032)

[Namespaces for test individuals 21](#_Toc398081033)

[Test Tools and Techniques 21](#_Toc398081034)

[The FIBO-OOPS!/OQuaRE Tables 21](#_Toc398081035)

[OQuaRE Assessment Criteria relevant to FIBO BCO 22](#_Toc398081036)

[Test data QA: OntoQA 23](#_Toc398081037)

[References 23](#_Toc398081038)

[Annex 1: Example Business Application Areas 24](#_Toc398081039)

[Foundations (FND) 24](#_Toc398081040)

[FND: Industry Business Areas 24](#_Toc398081041)

[FND: Other FIBO Support 24](#_Toc398081042)

[Business Entities (BE) 24](#_Toc398081043)

[BE: Industry Business Areas 24](#_Toc398081044)

[BE: Regulatory 24](#_Toc398081045)

[BE: Other FIBO 24](#_Toc398081046)

[BE: Interoperability 25](#_Toc398081047)

[Indices and Indicators (IND) 25](#_Toc398081048)

[IND: Industry Business Areas 25](#_Toc398081049)

[IND: Regulatory 25](#_Toc398081050)

[IND: Other FIBO Support 25](#_Toc398081051)

[Securities (SEC); also Equities, Bonds 26](#_Toc398081052)

[SEC: Industry Business Areas 26](#_Toc398081053)

[Debt – specific user stories 27](#_Toc398081054)

[SEC: Known Regulatory Requirements 27](#_Toc398081055)

[SEC: Other FIBO Support 27](#_Toc398081056)

[Derivatives (DER) 28](#_Toc398081057)

[DER: Industry Business Areas 28](#_Toc398081058)

[DER: Regulatory 28](#_Toc398081059)

[DER: Other FIBO Support 28](#_Toc398081060)

[Loans (LOAN) 28](#_Toc398081061)

[LOAN: Industry Business Areas 28](#_Toc398081062)

[LOAN: Regulatory 29](#_Toc398081063)

[LOAN: Other FIBO Support 29](#_Toc398081064)

[Phase II: Temporally Defined Terms (Price / Yields / Analytics): 29](#_Toc398081065)

# Introduction

Quality/Testing Requirements as stated in the main “Build / Test / Deploy / Maintain” document [1] are:

FIBO will be used as a reference throughout the industry for best practices. This means that FIBO publications will be held to a high standard of scrutiny for quality. A job of the FIBO Leadership Team (FLT) is to enforce high standards of quality for the published models, ontologies and specification documents including strict configuration management.

FIBO development will be done in any of a wide variety of tools, but publication following strict EDMC and OMG guidelines. These guidelines are well documented, and, to the extent possible, supported by automated tools. The FIBO team is able to determine quickly and accurately whether a submission satisfies the FIBO quality requirements through the use of an open source utility known as Jenkins, which is described in some detail below.

This document sets out how this is to be achieved. The Test Strategy is derived from first principles in quality and test, namely the distinction between verification and validation, in the senses used for those words in the testing literature. Many techniques exist for ontology evaluation and these are referenced here. This Test Strategy aims to make best use of these techniques and the published experiences of others in ontology evaluation, while automating as much of this as is practicable (and document what is to be done manually) to fit in with the overall Build / Test / Deploy /Maintain methodology.

# Test Strategy Principles

As set out in detail in the testing theory and references document [2], a test strategy embodies the following principles:

1. Verification
2. Validation

These are normally described with reference to programs, as follows:

* **Verification:** how well the program meets its formal specification(s)
* **Validation:** how well the program (and its specification) meets the stated business purposes.

Verification is typically covered in unit tests while validation is covered in system tests, integration tests or user acceptance tests. FIBO adapts this in that it is a structural artifact and is a conceptual model.

## Applying to Ontology

### The Two Vs

For an ontology, the two Vs translate to:

* **Verification:** tests with OWL Individuals framed according to the ontology itself
* **Validation:** tests against representative industry data

# Validation and Verification Implementation

Applying the principles of validation and verification, we have:

## Verification

Tests on physical ontologies, which return expected results.

These verify that a physical application, consisting of the ontology plus some datastore and some reasoning, querying etc. infrastructure, works as expected.

**Test data:** data which is framed according to the ontology itself will verify that the ontology is consistent, but do nothing to validate that the classes and properties in the ontology correspond to real world things.

## Validation

Identifying that the concepts presented in the ontology correspond to things in the domain of discourse, are modeled consistently, include the full scope of the intended concepts and so on.

Validation is often achieved by inspection or (for systems) by the completion of system integration tests.

**Test data:** data which is sourced from existing industry data, translated into suitable OWL individuals and data facts, will validate that the concepts in the ontology can correspond to the things which representative data sources describe.

# Test Data Considerations

## Verification Tests

Test data is in the form of OWL Individuals and facts (instances of properties) about those individuals.

These tests verify that the ontology is consistent.

**Example:** Tests on the submitted FIBO-FND and FIBO-BE against test individuals for Control will indicate whether concepts which are framed as “Role” types will be incorrectly classified by the reasoner.

These tests do not validate that the concepts in the ontology are meaningful. For example the test individual “BigBankPIR” as a kind of “PartyInRole” does not say anything about whether this instance corresponds to anything meaningful or represents any kind of real world data. If BigBankPIR is an individual of the class “PartyInRole”, verification tests don’t address the question of whether there is actually a use case in which we know that BigBank play some role in some context but we don’t know what it is and would need to run the reasoner to find out.

A more realistic test individual would be “BigBankAsSubsidiary”. When run against the reasoner, this test individual may be re-classified as a more specific sub-type of the initial concept, such as “Wholly owned subsidiary” or “Partially owned subsidiary”. Determining whether the concept is meaningful comes under validation.

These consistency tests should precede the Validation checks, however we should make the regression test individuals correspond to realistic concepts as validated under the “Validation” section.

### Unit Tests and Regression Tests

There will be a suite of tests which are run on every changed version of an ontology, in a unit testing environment. These ensure that something which worked before, still works. These are regression tests.

Regression tests are automated, but the first time a given test is run it typically would not be automated. The first time a test is run there needs to be some active evaluation of the results, which is not needed when the same test is run subsequently on later iterations of the ontology.

### Regression Tests: Unit Test

The suite of regression tests will be extended on the completion of unit tests of any new ontology or extension of an existing ontology. Changes and extensions to an existing ontology will require re-evaluation against the existing unit test level regression tests.

## Validation Tests and Checks

These are based on example real world data, and on test data individuals which have been framed with reference to the data structures of such real world data.

Validation tests not only the ontology but also the accompanying specification. We must validate not only that realistic data yields realistic and meaningful results when reasoned over or queried, but also that the information available in the specification provides enough information that someone is able to take real world data and map it to the appropriate concepts in the ontology.

These tests should also include testing what happens when someone misapplies the business conceptual semantics when mapping the test data. Does the ontology contain enough meaningful concepts that incorrect mappings are reliably identified. For this we should include some “bad” mappings in the test data.

Example: framing two PartyInRole concepts as the same concept when in fact they are different roles performed by the same individual (a common error!)

These tests would be against the conceptual ontology.

Identification of such errors would typically require more assertions about classes in the ontology than would be needed in any operational ontology, and may therefore extend the required scope of the ontology content.

### Validation Test Data

Original data should be provided in the form of tables, without reference to the ontology.

The data in these tables should be mapped to what are assumed to be the appropriate OWL individuals in the ontology.

The following kinds of exception should be trapped at this point:

* Test data literals which cannot be framed in terms of the datatypes provided in the ontology
* Relative things:
  + Is it clear from the FIBO documentation when to model something as a relative thing (PartyInRole, Asset, Underlying etc.) and when the data is descriptive of the thing without regard to context?
  + Is the context of the supplied data clear enough that the appropriate Mediating Thing can be populated in the test data as a business context or use case (e.g. clearing and settlement, decision support, risk)?
* Has data been mapped to the appropriate concepts in the ontology?

**Example:** given that applications frequently use “Country” as a surrogate for “Jurisdiction”, has the right ontology class been chosen?

Then: given a piece of test data called “LondonJurisdiction”, populate the additional classes to which this has a relationship, then present the results back for inspection. What is the range of hasReach and is it something (a) meaningful and (b) true to the facts on the ground (London is in the jurisdiction of EnglandAndWales, which has reach England and hasReach Wales).

This step requires inspection. For the above example, it should be apparent that the range of ‘hasJurisdiction’ for something in London is in fact the ‘EnglandAndWalesJurisdiction’. Any attempt to create an individual called ‘LondonJurisdiction’ should be identifiably incorrect.

These inspections help to ensure that the ontology provides sufficient context to disambiguate otherwise similar concepts. For example, does the ontology support the distinction between a Loan Product, a Loan Contract and a credit Facility?

This is the case even if some of those disambiguation classes are not needed in the operational ontology (FOO); they are there to allow for validation of meanings, which in turn provides interoperability across disparate applications.

### Regression Tests: Semantic Enhancements Validation

Enhancements of an existing ontology for a given set of concepts will require re-running of existing tests.

Example: Suppose the model is refactored so that Contract becomes a sub class of Agreement where once it was a separate concept corresponding to the written (including virtual) contract. Subsequent to this refactoring it will be necessary to run the same tests as were run before, to ensure the same results are achieved based on the same information.

In the above example, tests after refactoring clearly require new “Validation” tests, since the source data would be expected to be mapped to different classes and properties, and we would need validate that this mapping can be carried out based on the information available. This means that some regression test s will take the form of a repeat of some validation test, parts of which may or may not be automated.

# Test Strategy

Per Build / Test / Deploy /Maintain document [1] under “FIBO Content Teams”:

* If a UML model was selected, convert to RDF/OWL, else run the modified RDF/OWL through QC (OOPS, OQUARE, etc. as Jenkins jobs)
  + Consistency checks
  + Syntax validation
  + Reasoners
  + Regression tests

### Use Case of the ontology itself

There are three possible styles of ontology

* ***(1) Fully Conceptual (domain reference) ontology:*** no technology constraints
  + These are outside the scope of this test strategy
  + In practice most of these are also DL-safe but this is expressly not a requirement
  + These may also contain inverses which are not seen as needed in operationally-ready ontologies for performance reasons but would be needed for other e.g. Querying use cases, linked data semantic and so on
* ***(2) Operationally ready ontologies:*** constrained to DL-safe operation
  + **User story:** user is an ontologist wishing to create a reasoner-based or SPARQL querying application
  + **Use case:** users shall be able to stand up operational ontologies by direct use of the ontology or copying it into a separate namespace for use of a sub-set of the whole
  + **Use Case:** users shall be able to map terms to and from this ontology for conventional datastores, message feeds and other data resources
  + Other user stories
    - Not all industry applications will require ontologies to be constrained to this level – see Fully Conceptual Ontology to meet other requirements
    - This is currently the assumed default for OMG submissions though this remains at the discretion of PCT leads
* ***(3) Operational Ontologies:*** constrained to the performance requirements of an individual application which is expressly designed to meet an individual use case or defined set of use cases.
  + These are beyond the scope of this Test Strategy

HOWEVER

* + The main use case for the Operationally Ready ontologies in (2) above is that for each user story for which users may want to use a given FIBO ontology, that ontology shall be able to provide the concepts needed
  + Operational ontologies use cases therefore provide one of the main guiding principles for the scoping and the satisfaction of user stories for the Operationally ready ontologies covered by this Test Strategy.

These tests are for ontologies which are intended for the use case of operationally-ready ontologies and does not apply to FIBO Red / fully conceptual ontologies which are necessarily unconstrained.

Note “FIBO-OOPS!/OQuaRE” below refers to the FIBO adaptation of the OQuaRE / OOPS! combined table as described in [2]and given in [3].

## Verification Measures (Unit Test)

### TBox Tests

*TBox is the term for the ontology of classes and properties which define those classes.*

On the TBox (ontology itself):

* Consistency checks
* Syntax validation
* DL-safe validation
* Run selected FIBO-OOPS!/OQuaRE measures as identified for this stage of testing
  + Pass / fail measures
  + Numeric measures (e.g. clustering of concepts, depth of subsumption hierarchy);
    - for these record the measure and compare with other ontologies

### ABox tests

*ABox is the term for a set of individuals with facts about those individuals*

* Populate the chosen ontology with OWL Individuals and facts (instances of properties) for all terms in the ontology (at least one example individual per class in the ontology);
  + In later regression tests this new set of individuals will be re-used as the regression test set
* Run basic sanitary checks on the resulting combination of TBox and ABox ontologies
  + FIBO-OOPS!/OQuaRE measures as appropriate for ABox tests
  + Other tests (OntoClean etc.) as applicable
    - [this part of the Test Strategy to be fleshed out with specific evaluation measures as tests progress and we find out what works]

## Validation Tests (analogous to system and integration tests)

* **Test Objective:** Validate that the ontology reflects the real world meaning of the subject matter
* **Test Objective:** Validate that the ontology provides the semantics needed for a representative set of use cases or user stories in the set of business areas / business processes / industry workflows (intra-firm and cross-industry workflows) for which the FIBO ontologies under test are intended to provide semantics
* **Test Objective:** Validate that the ontology and its accompanying guidance information provides sufficient information for it to be populated accurately with ABox data and such that the results of that populated material yield meaningful results for queries
* **Test Objective:** Validate that the ontology may be mapped to and from unstructured data sources and data sources in other than triple-store formats
* **Test Objective:** Validate that the addition of restrictions during refactoring yields business results which are in line with the originally intended semantics of the concepts
* **Test Objective:** Validate the semantic enhancements carried out during refactoring of the ontology set from the legacy models

These break down as follows:

### Semantics and Sensibility

* **Test Objective:** Validate that the ontology reflects the real world meaning of the subject matter

*1. Checks for sensibility of concepts*

Using an ontology editor, create OWL individuals for the classes in the ontology.

For each class, create facts (instances of properties) for each property which has the class as a range and each property which is the onProperty target of a restriction on that class

For each object property in the above, create an individual for each class which is the range of that property. Include at least some individuals which are sub classes of that class.

Produce a graph (e.g. via a SPARQL query) of those individuals and their relationships. Represent this in one of the business facing formats specified as valid FIBO business facing presentation formats (non-technical diagrams, tables or controlled natural language).

Subjective assessment:

* Do the statements created by the graph make sense to the modeler
* Present these statements to a subject matter expert: do these statements reflect the reality of those terms?

Example (class names capitalized for identification):

**Example:** *“London is in the London Jurisdiction, which has reach over the United Kingdom”* is clearly untrue and shows that the class “LondonJurisdiction” was wrongly populated with a meaningless individual; did the ontology or the associated guidance notes allow the tester to make this mistake?

(if possible this test should be carried out by SMEs with limited ontology knowledge)

**Example:** *“London is in the England and Wales Jurisdiction which has reach over England and has reach over Wales”* is a true and meaningful statement.

**What is assessed:** does the ontology cover enough properties and enough classes which are ranges of those properties, that a graph can be created which defines sufficient detail for the main subject-matter classes to be clearly disambiguated from other, similar concepts?

Note: This test requires more classes (as ranges of properties) than may be needed for individual use cases. It validates the fully conceptual / domain ontology use case of disambiguation of concepts in a given business area. Provided concepts can be fully disambiguated with the conceptual reference ontology, not all these classes need to be stood up in operationally ready ontologies.

*2. Use of Relativity Partitions*

**Note:** this evaluation method uses the conceptual reference ontology, and is used to ensure that the corresponding terms in the operationally ready ontologies under test are well-formed semantically with reference to the upper ontology.

**Note 2:** It may be best to combine this with one of the mapping validation evaluation activities shown below. This would combine the two requirements of (1) proving that data can be mapped sensibly to the ontology both for independent and relative things, and (2) proving that errors in doing so are reliably trapped by the ontology.

Populate the ontology with one or more individuals for each class which is intended to be an “Independent Thing” and for each class which is intended to be a “Relative Thing” (such as a party in a role, an asset, a derivative underlying, an economic resource etc.)

Carry out this test in the conceptual reference ontology, either

* Directly on the conceptual reference ontology, or
* by using the upper ontology lattice as a kind of test harness
  + For example add an owlImports relation to the Upper Ontology / Relativity lattice from the ontology under test, in the test namespace and making each class in the ontology under test, a sub-class of the appropriate partition.
  + Use a version of the Upper Ontology or RelativityLattice ontology which has disjoints applied between independent, relative and mediating things.
* Then:

Run the reasoner.

*Expected Result:* After running the reasoner, no class which was defined as a kind of independent thing shall be reclassified as a relative thing, and no class which is a relative thing shall be reclassified as an independent thing. Also no class of either of the above shall be reclassified as a Mediating Thing.

*Combining with Mapping Validation:*

* A “fail” in the above may result in opposing results (Fail or Pass) for the ontology itself as follows:
  + If the concepts are clear and correct and the reasoner causes misclassification, record a fail for the ontology
  + If the concepts have been misapplied (as determined from inspection, third party review etc.) and the ontology causes this to misclassify, this is a Pass for the ontology itself: the information contained within the ontology has (with the addition of the upper ontology test harness) provided enough semantics to detect and prevent this.
    - At the same time this is a fail for the supporting documentation.

*3. Use of other partitions*

To follow later – for example correct application of temporality treatments; segregation of pricing etc. from specifications of those. These tests would be formed similarly to the above and would use the other Upper Ontology partitions in the conceptual reference ontology, either as a test harness or by testing directly in the conceptual reference ontology and then extracting the subset of this which is to be taken forward.

### User Stories and Use Cases

* **Test Objective:** Validate that the ontology provides the semantics needed for a representative set of use cases or user stories in the set of business areas / business processes / industry workflows (intra-firm and cross-industry workflows) for which the FIBO ontologies under test are intended to provide semantics

These tests are about ensuring that a representative range of user stories can be supported by the ontology. Note that this mainly determines the scope both in terms of range and of depth (see theory document) but that scope maybe broader if there is the additional use case of interoperability with other ontological standards in this or a related business area (deeper abstraction to support interoperability, than may be needed for any of the individual use cases).

Identify and itemize a list of business areas. For example, interactions between business actors which result in the exchange of or reference to some information (trade confirmation, reporting etc.) or the use of business information for risk assessment, investment decision support and the like. Break these down into user stories (for example the workflow of each interaction between the actors in that space) and to the extent possible break these down into detailed use cases (as use case tables, supported by UML Use Case models for diagramming of those).

#### GOEF Methodology

Here we apply the GOEF methodology to those use cases:

Split the use case into:

1. Functional Objective
2. Design objective and requirements specification
3. Semantic components required to achieve above

[Editor note (MGB): see GOEF literature for more details of this methodology. I’ve teased out some points below – these are not based in the GOEF literature but on my interpretation of this in the light of what we need to validate for FIBO].

*(1) “Functional Objective”:* what goal is the actor trying to reach? [check I have this right!]

*(2)* For the *“Design objective”* part, identify the architecture or range of possible storage formats in which the information needed for the functional objective is likely to be (or known to be) framed in the application.

Architectures:

* existing applications are likely to be known and documented, and would typically be in conventional technology architectures
* some use cases will identify new business activities / user stories which can be met with semantic technology applications, or use cases which are currently met by conventional applications (like risk and compliance) which would be better met by reasoning or semantic querying applications using triple stores

List out the data storage formats to be supported.

*(3) Semantic Components:* what are the concepts for which actors in the activities described need to receive, learn or understand their values? That is, given that the data gives the truth values of propositions, what are the semantics of those propositions? [again I may be way off beam here]

Put simply: What meanings are needed in this application? What is the information which needs to change hands in the workflow, or which needs to be inspected by some end user? For these kinds of information, what are the meanings of the concepts?

This determines the required scope of the ontology if it is to support this use case.

**Recall:** applications *meet* a use case; FIBO as a conceptual model for the application must *support* the use case – not the same thing! This makes a subtle difference to the tests compared to what there would be if these tests were a test of an application (application *meets* use case).

Scoping of a set of FIBO modules and ontologies should have been based on a set of such user stories. For each of the user stories identified during scoping, these validation tests are to be carried out.

### Population from semi-structured Data

* **Test Objective:** Validate that the ontology and its accompanying guidance information provides sufficient information for it to be populated accurately with ABox data and such that the results of that populated material yield meaningful results for queries

Carried out alongside the next Test Objective:

* **Test Objective:** Validate that the ontology may be mapped to and from unstructured data sources and data sources in other than triple-store formats

For each of the use cases listed from the previous section, create ABox material (individuals) based on real world data which corresponds to those use cases.

This data shall not originate in Triple store / RDF formats but should be representative, ideally tabular data, as seen in reports of, or query results on, the data which is actually used in those applications (for existing applications) or (in the case of use cases for new semantic technology applications to be developed) data set out in tabular or spreadsheet form by representative business actors in the processes to be supported.

Why? The aim here is that, for each user story and / or use case, and for the concepts needed to support those as determined from the GOEF analysis in the preceding section, it is possible for the ontology to (a) represent all those concepts and (b) do so accurately and meaningfully. At the same time, the above validation test objective is covered: that anyone can put the right data in the right boxes, as it were. That is, that the OWL individuals may be stood up correctly following the advice given, and it becomes readily clear from the model semantics if they are not.

*This is the main Validation activity: verification unit tests identified whether the ontology was consistent but did not say if the meanings were right; these tests try to determine whether the meaning is right.*

There are two or more practical methods to achieve this Test Objective. Use the tabular or unstructured data to create individuals and facts, and then:

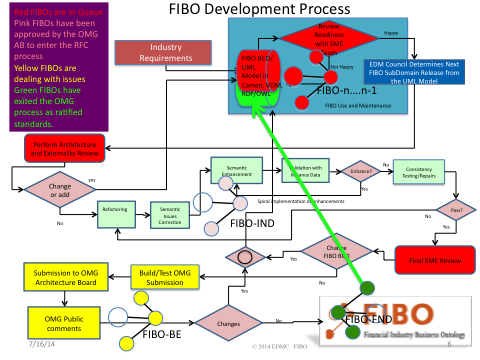
* Create a graph of the model content or a set of this, via a SPARQL query, and set this in front of an SME in one of the recognized business formats and ask if the end result is meaningful
* Make a list of competency questions which the actor needs to be able to answer in order to carry out the activity described for them in the user story or use case,
  + Frame each of these as a SPARQL query
  + Interrogate the ontology using this query
  + Determine if the required answer is meaningful to the actor (via inspection by a representative SME), or
  + Determine if the required answer is true, given the truth of the concepts which were queried

Competency questions and their required answers, having “Passed” once on an ontology, shall subsequently become **regression tests** and shall be automated along with their required answers for the test data set which was used.

### Refactoring Validation Tests

The next 2 Test Objectives follow from the activities in the “refactoring” detailed workflow (figure 1)

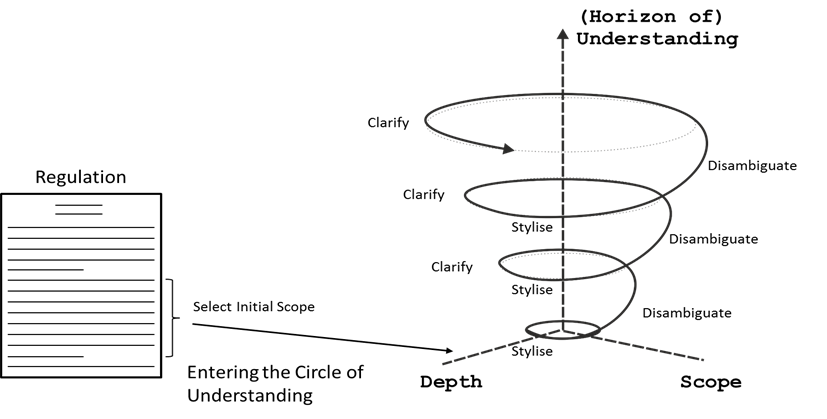
* **Test Objective:** Validate that the addition of restrictions during refactoring yields business results which are in line with the originally intended semantics of the concepts
* **Test Objective:** Validate the semantic enhancements carried out during refactoring of the ontology set from the legacy models



**BDD activities**

**Figure 1: Build / Test / Deploy / Maintain Process Flow. Elements tested here are circled in blue**

Although this is an ontology, both of these activities are highly design-like in nature, and so each step in the design should advance according to the “Behavior Driven Development” spiral process as discussed in Section 4.4. of the Build / Test / Deploy / Maintain document [1]. See Figure 2 (also from [1])



**Figure 2: Spiral Continuous Integration Methodology, with testing points highlighted**

In order to support the notion of a spiral, each change introduced in these two activities must represent an incremental change on the existing legacy model, which forms the “core” to which these changes are made. Further, it makes to do the simpler activities before the more complex ones.

Here, the simpler activity is simply replacing each stand-alone object property in the legacy models, with restrictions. This formalizes the logic behind the semantics without changing the intended semantic of at all: the legacy models assume that the existence of a property for a class equates to this being a necessary condition for membership of that class, unless the cardinality is 0. This can be easily tested. The more complex activity is the addition of complex chains of restrictions or of restrictions plus unions, and the joining together in the ontology of properties or restrictions which were asserted quite independently in the legacy model. Those are what I defined as “semantic enhancement” since the nature of this activity is to try and say more with the expressive power we have, than was attempted in the legacy models. This is where things can go wrong.

This section therefore describes the development process as well as the test process for this activity, since they are inextricably linked.

#### a) Refactoring: Properties Replacement with Restrictions

* **Test Objective:** Validate that the addition of restrictions during refactoring yields business results which are in line with the originally intended semantics of the concepts

For each property in the legacy model:

* Determine whether it introduces a new meaning to the model or is interpretable as the refinement or reuse of the meaning of an existing property
* If this can be framed as the refinement of an existing property, replace the old property with a restriction on that property (this is generally the parent property or an ancestor property of the one being replaced).
  + Example: “Bond has interest payment terms” might or might not be a reuse of the property “Debt has interest payment terms”
  + Simple verbal validation with SME: read out the name of the domain followed by the name and /or the written definition of the property being reused followed by the range”: does this seem to make sense? If not, reverse out the change, reinstate the original sub-property and try again or leave it as a unique concept.
* Generate a graph of the resultant ontology in a business-facing format (including business language representation of the restriction, e.g. styled as a reuse or refinement of a property).
  + Present this to SMEs – does the semantics of the resultant model look correct?
  + Compare with the legacy model diagram and with any session notes, diagram notes etc. which would shed light on the original intended meanings: is the resultant model consistent with that?
* Populate this part of the ontology with representative test data (from regression tests if available. Or create as new, as part of the earlier mapping tests).
* Run regression tests, including consistency checks, other FIBO-OOPS!/OQuaRE tests and measures, ensure before and after results match (there should have been no change to the semantics at this point)

#### (b) Refactoring: Semantic Enhancements

This is the addition of complex chains and sequences of restrictions, combinations of restrictions and unions, and so forth.

These are changes which aim to make better use of the expressive power of the ontology language than was done at the time of the legacy models SME reviews. Strictly these are not issues with the original models but attempts to say more. Therefore anything more which is said should be validated. These changes also have to be verified i.e. proven to be consistent, not cause reaoning errors of misclassification and so on.

Also numeric measures in the FIBO-OOPS!/OQuaRE test suite should remain within the bounds that have been determined as acceptable for each such measure.

These are often done alongside corrections to errors found or believed to have been found in the legacy models, so these are treated as the same thing for testing purposes.

These enhancements should be approached with caution: assertions in the legacy model are often segregated for a reason. It is tempting, with a “design” hat on, to try to apply knowledge of “what follows what” according to one’s knowledge of the world. However, conflating concepts in the ontology with conclusions that would follow from reasoning or other processing can only degrade the ability of the ontology to provide both the inputs and the output semantics of a wide range of queries and reasoning applications.

For example just because holding of a 51% stake in a company usually means that one “controls” that company, does not mean these concepts should be conflated in the ontology – the concepts of holding an interest and having some control are semantically distinct even if the truth value of one were always to follow from the truth value of the other. Which it doesn’t. Many regulations require even further refinements of both of these concepts separately for precisely these reasons (for example one regulation may treat 25% as a controlling stake for the purposes of that regulation simply to get around the fact that they don’t always follow) – and the use cases represented by those regulations would then not be met by the ontology design.

This part of the testing therefore needs to go right back to the use cases and user stories.

Any tendency of an ontology designer to conflate meaning with truth and try to create hidden inferences within the ontology which are actually not supported by the business meanings, should be met with force.

So:

For each use case: for each proposed enhancement to the ontology, determine how it would affect that use case.

#### Overview / Nature of These Tests:

In order to take care of all of the above, we need to:

* Create a baseline set of tests as described in the preceding sections – for a version of the legacy model in which only direct replacement of properties with restrictions has taken place, and has been tested;
* Suggested innovations may come about through one of two means:
  + Suggestions made by the modeler when looking at the model i.e. opportunistic improvements;
  + Drive by specific use cases: somewhere in the set of use cases identified so far, in the “semantic components” part of the GOEF output, there is a concept which is not being met by the current ontology and can be met by creating some new combination of model restrictions and other elements.

NOTE: The nature of this “Behavior Driven Development” or user story / agile development method requires only the second of those to happen. Development to date has resulted only from the first.

As a matter of principle, the addition of semantic enhancements which say more than was said in the legacy models should only be done in response to a known use case of a known application of FIBO in the semantic technology space.

**Test Process:**

* The first turn of the spiral takes the model with only replacement of properties with restrictions as required (the output of the previous stage of development and test);
* For each use case in the identified use case set:
  + Identify each of the “semantic components” as derived from the GOEF analysis;
  + Match these against concepts already in the ontology at this turn of the spiral;
  + Identify any missing concepts; these will either be:
    - Things that require higher orders of logic in order to state them
      * Record these for later enhancement using RIF rules in the FIBO-RIF dialect – these are to be dealt with later;
    - Things that can be stated using some combination of restrictions and unions

Note:

* over time we should maintain a library of combinations of restrictions and unions which have passed previous tests in comparable situations (these are what are sometimes referred to as ontology design patterns in one of two senses of that term, the other being archetypical semantic abstractions – see Ontology Summit 2014 Track references in [2[Appendix 1).
* We should also maintain a corresponding set of business facing representations of those patterns which are proven to work
* Design a set of restrictions patterns whose outcome is intended to be the missing semantics
* Carry out tests as follows:
  + Basic verification tests at the top of this section
  + FIBO-OOPS!/OQuaRE tests as identified previously
  + Regression tests – all regression tests that have been identified to date for this ontology
    - For second and subsequent turns of the spiral, include as regression tests those tests which were performed on previous turns of the spiral
  + ABox testing using sample data for the “semantic components” in question along with all sample data used to date in the regression tests
    - To ensure that the new meaning is supported
    - To ensure that the meanings already assumed to hold for the affected items still hold true.
  + Queries based on competency questions that are derived from the use case for which the enhancement was made.
* Rinse and repeat: each use case or user story results in one more turn of the spiral

# Test Namespace Requirements

We need:

* Namespaces for FIBO Ontologies
* Namespace for Test Ontologies
* Namespaces for test individuals

## Namespaces for the Ontologies

At present these are:

Elements excluded for OMG operational constraints would be included in extensions as before; Legacy OWL does not identify these

[www.edmcouncil.org/spec/EDMC-FIBO/XXX/Abcde/Fghij/](http://www.edmcouncil.org/spec/EDMC-FIBO/XXX/Abcde/Fghij/)

Where XXX = Corresponding Spec level mnemonic (e.g. IND), with non OMG Module and / or ontology not included in OMG submission (inverses etc.)

Items not yet submitted (Red / Pink):

[www.omg.org/spec/EDMC-FIBO/YYY](http://www.omg.org/spec/EDMC-FIBO/YYY)

Where YYY = OMG Spec namespace element

[www.edmcouncil.org/spec/EDMC-FIBO/XXX](http://www.edmcouncil.org/spec/EDMC-SR/XXX)

Where XXX = OMG Spec namespace element e.g. IND

Surrogate for published ontology URI which is:

[www.omg.org/spec/EDMC-FIBO/XXX](http://www.omg.org/spec/EDMC-FIBO/XXX)

[www.edmcouncil.org/spec/EDMC-FIBO/FND/](http://www.edmcouncil.org/spec/EDMC-SR/FND/)

[www.edmcouncil.org/spec/EDMC-FIBO/UO/](http://www.edmcouncil.org/spec/EDMC-SR/UO/)

[details to be agreed and updated here – Mike, Dean, Jacobus]

[www.w3.org/2000/01/rdf-schema#](http://www.w3.org/2000/01/rdf-schema) ; [www.w3.org/2002/07/owl#](http://www.w3.org/2002/07/owl) ; [www.w3.org/2001/XMLSchema#](http://www.w3.org/2001/XMLSchema)

## Namespace for Test Ontologies

Copied from ongoing namespace conversations:

* We want to distinguish the various color-versions of FIBO in the URI
* We want potentially resolvable URIs that are
* compliant with how Protégé and TopBraid load external imports
* compliant with the Linked Data standard (for the ABox elements at least)
* The host name in the namespace URIs should initially be \*.[edmcouncil.org](http://edmcouncil.org/) and later \*.[omg.org](http://omg.org/)

|  |  |
| --- | --- |
| **Stage** | **Namespace** |
| Pink | [http://ontology.edmcouncil.org/fibo/pink/fnd.owl#](http://ontology.edmcouncil.org/fibo/pink/fnd.owl) |
| Red | [http://ontology.edmcouncil.org/fibo/red/fnd.owl#](http://ontology.edmcouncil.org/fibo/red/fnd.owl) |
| Yellow | [http://ontology.edmcouncil.org/fibo/yellow/fnd.owl#](http://ontology.edmcouncil.org/fibo/yellow/fnd.owl) |
| Green | [http://ontology.omg.org/fibo/fnd.owl#](http://ontology.omg.org/fibo/fnd.owl) |

*Was this meant to be the test namespacs or the Red /Pink/Yellow / Green namespaces? If the former, the OMG one is wrong.*

## Namespaces for test individuals

These should identify the test activity and / or the use case or user story which they represent. Some test ontologies may cover more than one requirement.

Note that these ontologies of test individuals should be tested using OntoQA for coverage against the TBox ontology to which they refer.

# Test Tools and Techniques

## The FIBO-OOPS!/OQuaRE Tables

A table has been created for a FIBO implementation of OQuaRE [3]. The existing table is quite complex – however we should be able to frame a number of tests and other measures which satisfy multiple evaluation criterial, giving us a fairly simple regression test suite.

A column in the tables indicates which evaluation criteria apply to Conceptual, to Operational or both.

Measures in OQuaRE and OOPS! include objective measures of the ontology such as depth of subsumption hierarchies, clustering of concepts and so on, many of which are not a pass/fail but a numeric measure of something about the ontology. For these measures we must identify what we want the results of those measurements to look like.

Sometimes a numeric measure of the ontology may have a different requirement for operational and for conceptual ontologies. That is, these tests measure the trade-offs between expressivity and tractability in reasoners. These numbers will need to be recorded.

### OQuaRE Assessment Criteria relevant to FIBO BCO

After deleting the criteria from OQuaRE which are not relevant to the BCO, we have:

* Structural
  + Formalization
  + Formal Relations support
  + Cohesion
  + Tangledness
  + Redundancy
  + Consistency
  + Cycles
  + Structural Accuracy
  + Domain Coverage
* Operability
  + Appropriateness
  + Recognizability
  + Learnability
  + Ease of Use
  + Helpfulness
* Maintainability
  + Modularity
  + Reusability
  + Analyzability
  + Changeability
  + Modification Stability
  + Testability
* Functional Adequacy
  + Reference ontology
  + Controlled vocabulary
  + Schema and Value Reconciliation
  + Consistent search and query
  + Knowledge Acquisition
  + Clustering and Similarity
  + Indexing and linking
  + Results representation
  + Classifying instances
  + Text analysis
  + Guidance and decision trees
  + Knowledge reuse
  + Inferencing
  + Precision

## Test data QA: OntoQA

This provides measures of a knowledge base. The measures here can be applied to the A-box instance data files which make up the test applications.

OntoQA gives a coverage metric, which should be applied to the test data set to measure how much of the ontology it covers.

# References

1. “FIBO™ Build, Test, Deployment and Maintenance Environment”, filed as [date]\_FIBO build test deploy maintain.docx –working document (current ref is to 20140908 version)
2. “FIBO Test Strategy Theory and References”, filed as FIBO Test Strategy Theory.docx on GitHub
3. FIBO OQuaRE Table – see document OQuaRE FIBO Table v1.docx on GitHub (where v1 is the current version)

See Reference [2] for further references.

# Annex 1: Example Business Application Areas

In each case:

* Activities carried out in the financial industry which the ontologies are to support
* Regulatory Requirements / application areas
* Other FIBOs which are to be supported

Also noted e.g. in BE:

* Requirements for interoperability with semantics in other resources

## Foundations (FND)

### FND: Industry Business Areas

Not applicable

### FND: Other FIBO Support

All FIBOs

## Business Entities (BE)

### BE: Industry Business Areas

There are numerous applications of FIBO Business Entities beyond the merely financial. We should focus on the financial industry use cases for formal validation, but also allow for interoperability by ensuring we have adequate levels of abstraction as needed to map to other ontologies and industry data models, and validate that this need is met (or record if it is not).

### BE: Regulatory

* Regulation W
* Anti money laundering
* Know Your Customer (KYC)
* Insider trading and other conflicts of interest
  + Regulation W
  + Accredited Investors (Regulation S / 144A etc.)
* Counterparty exposure: transitive exposure across parent / subsidiary relations
* Tax: FATCA

### BE: Other FIBO

* Service providers /service provision abstractions, to support:
  + Exchanges and MTFs
  + Credit facilities
* Structured Finance (SPVs)
* Securities: reference data around issuers, guarantors, underwriters etc.
* Corporate Actions and Events (CAE)

### BE: Interoperability

Depth of abstraction will determine interoperability. Given the near-universal nature of business entities concepts, we should consider interoperability. Possible interoperability requirements may include:

* Linked Open Data –common point of reference on entities information
* Rights (contracts, IP, transactions, logistics etc.)
* Security

## Indices and Indicators (IND)

### IND: Industry Business Areas

**Treasury**

* Corporate Finance (including foreign exchange etc.)
* Working Capital Management

**Front Office / Decision Support (fund management):**

* Cash balance management
* What-if analysis
* Performance attribution

**Central Bank**

* Statistical Analysis
* Capital adequacy
* Currency amounts in circulation (macroprudential analysis)

### IND: Regulatory

* Research on treasury regulatory requirements”
  + Possible role in price transparency
* Macroprudential and microprudential oversight (currency, exchange rates etc.)

### IND: Other FIBO Support

* Derivatives:
  + Interest Rate Derivatives
* Debt:
  + Floating Rate Notes (in Bonds)
  + Inflation debt instruments (in Bonds)
  + Some Amortizing Securities (in Structured Finance or in Bonds)
* Loans
  + All loans sections (variable interest repayment terms)

In practice most of the above will be captured in Common Debt Terms as they do not vary between Bond and Loan structures.

## Securities (SEC); also Equities, Bonds

The scope was determined by the ISO 20022 “FIBIM” model and the application areas given are the ones we know of that are supported by ISO 20022FIBIM, as well as by ISO20022 more generally, ISO 15022 etc. That model also embraced CIV (Funds) and certain Rights instruments (separate sections in the FIBO Reference ontology).

Securities Common Terms is intended to capture the abstractions common to all financial instruments, to all negotiable securities, to all exchange traded securities and so on, as an abstraction hierarchy of contract types. These abstractions won’t be encountered in most use cases but are intended to support disambiguation between these concepts. These cover common concepts like security identifiers, classifications, tax treatments, regulatory status (Reg A /144A etc.), issuers, countries and currencies of risk, markets on which traded and so on, in support of all of the use cases identified for equities, for debt instruments, for rights instruments, for tradable find instruments and so on.

### SEC: Industry Business Areas

**Front Office:**

* Front Office (fund management):
  + Securities Research (bonds, equities, structured finance etc.)
  + Decision support (portfolio management) excluding price /yield /analytics terms
  + Performance attribution (when price / yield / analytics is available)
  + What-if analysis (when price / yield / analytics is available)

**Back Office**

* Master Data Management
* On-boarding (population of SMF when a new security is bought)
* Accounts / Reconciliation
* Reporting

**Middle Office**

* Risk Assessment /risk Management
* Compliance – internal (e.g. ensuring conformance to stated fund strategy for managed funds
* Compliance – to regulatory requirements

**Transactions Workflow**

* Trade confirmation and matching (counterparties transmit similar messages to custodian, including enough information to reliably identify the trade and the instrument –certain key instrument terms needed for this)
* Reference data elements for all process flows in the securities transaction lifecycle:
  + Trade
  + Post-trade
  + Clearing
  + Settlement

**Other industry activities involving reference data semantics**

* Buy side to sell side / broker communication
* Broker to exchange
* Exchange to clearing house
* Clearing house to custodian banks
* Custodianship
* Exchange data (principally temporal data; some reference data concepts maintained)

**Central Bank**

* Statistical Analysis\*
* Capital adequacy
* Currency amounts in circulation (macroprudential analysis)
* Etc.

**Corporate Actions**

* Transmission of corporate events and actions data (CAE) (reference data component)
* Given the choice, the statistics department of a central bank will take whatever information can be said to exist, without limitation. This is not a reliable use case for scoping because they will want it all.

### Debt – specific user stories

* Centralized bond trades reporting (see ICMA)

### SEC: Known Regulatory Requirements

Which have a need for a securities reference data component:

* Capital Adequacy (CCAR etc.)
* Risk Data Aggregation (RDA) (Basel II or III?)
* Fair pricing (MiFID) /Price discovery
* Regulation S /144A
* Regulation W
* Transparency
* Etc.

### SEC: Other FIBO Support

* Derivatives
  + Asset derivatives
  + Index Derivatives
* CIV / Funds
* Pricing / Market Data
* Corporate Events and Actions
* Securities Issuance Process

## Derivatives (DER)

The scope was initially determined by the FpML specification, with additional instrument types identified by SMEs. Additional scope (derivatives cashflow terms) was determined by Nordea Bank.

### DER: Industry Business Areas

* Per FpML message context in the message work flow (same content, different context):
  + Trade confirmation
  + Trade reporting
* Centralized Swap Data Repositories
  + See details published by various regulator for swap data requirements
* Hedging
  + Hedging risks of other transactions- in conventional funds
  + Hedge funds
* This involves defining strategies in terms of combinations of derivative and tradable instruments, comparing analytics (yield curves) and the like –alongside Price/Yield/Analytics
* Exchange Traded Derivatives: marketing / trading (standardized derivatives terms offered on exchange)

### DER: Regulatory

* Price Comparison

**Note:** the ISDA scheme for classifying derivatives is based on this requirement; other classification facets would reflect other regulatory or business requirements, such as credit / risk exposure, which do not depend on the distinctions made in ISDA for Vanilla v Exotic, for example. This suggests that the user stories for derivatives will drive the scope and range of classification facets in the model.

### DER: Other FIBO Support

* CIV/Funds
* Portfolio

## Loans (LOAN)

Scope in FIBO was to be based on MISMO. Instead this model was scoped during IBM Research PoC activity, based on required inputs for risk models for structured finance.

Notably the scope introduced in FIBO and validated several times since (incl. GE Capital, ANZ) covers the range of “Credit Facility” concepts which are NOT in MISMO.

Therefore FIBO scope is broader than MISMO.

### LOAN: Industry Business Areas

* Lending
  + Retail lending
  + Commercial lending
* Credit Risk
* Capital Adequacy
* Reporting

### LOAN: Regulatory

* Wholesale Credit Reporting (FRY 14K? – check the name of this; see BAML mapping work)
* Others not researched

### LOAN: Other FIBO Support

* Structured Finance
  + Loan and mortgage loan information required for definition of structuredproducts
* Structure Finance risk management / assessment
  + Postcode area risk, other risk factors (borrower demographics, currency, industry sector etc.)

**User story:** mortgage lender who thought their lending was well distributed across different states in the US, when actually homes for which the funds were lent were all in Florida. The “borrower address” record was not coextensive with the address of the risk of the property itself. By assuming the borrower address “meant” the same thing as the collateral property address, the concentration of risk in one area was not detected.

## Phase II: Temporally Defined Terms (Price / Yields / Analytics):

Most of the use cases articulated above for reference data, will also require semantics for temporally defined data such as prices, yields, analytics etc.

**Notable additions to these requirements:**

* Exchanges: price reporting