Date: March 2013

Financial Industry Business Ontology – Foundation

Submission for RFC Comment

OMG Document Number: finance/2013-03-01

Standard document URL: http://www.omg.org/spec/FIBO/FIBO-Foundation/1.0/PDF

Associated File(s)*: as indicated in inventory file finance/2013-03-02 through 04

Source document: Financial Industry Business Ontology for Business Entities (finance-13-03-01)

* Original file(s): Financial Industry Business Ontology for Business Entities (finance-13-03-01)

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

OMG's Issue Reporting Procedure

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page http://www.omg.org, under Documents, Report a Bug/Issue (http://www.omg.org/technology/agreement.)

Table of Contents

Preface

About the Object Management Group

OMG

Founded in 1989, the Object Management Group, Inc. (OMG) is an open membership, not-for-profit computer industry standards consortium that produces and maintains computer industry specifications for interoperable, portable, and reusable enterprise applications in distributed, heterogeneous environments. Membership includes Information Technology vendors, end users, government agencies, and academia.

OMG member companies write, adopt, and maintain its specifications following a mature, open process. OMG's specifications implement the Model Driven Architecture® (MDA®), maximizing ROI through a full-lifecycle approach to enterprise integration that covers multiple operating systems, programming languages, middleware and networking infrastructures, and software development environments. OMG's specifications include: UML® (Unified Modeling Language™); CORBA® (Common Object Request Broker Architecture); CWM™ (Common Warehouse Metamodel); and industry-specific standards for dozens of vertical markets.

More information on the OMG is available at http://www.omg.org/.

OMG Specifications

As noted, OMG specifications address middleware, modeling and vertical domain frameworks. A Specifications Catalog is available from the OMG website at:

http://www.omg.org/technology/documents/spec_catalog.htm

Specifications within the Catalog are organized by the following categories:

OMG Modeling Specifications

- UML
- MOF
- XMI
- CWM
- Profile specifications

OMG Middleware Specifications

- CORBA/IIOP
- IDL/Language Mappings
- Specialized CORBA specifications
- CORBA Component Model (CCM)

Platform Specific Model and Interface Specifications

- CORBAservices
- CORBAfacilities
- OMG Domain specifications
- OMG Embedded Intelligence specifications
- OMG Security specifications

All of OMG's formal specifications may be downloaded without charge from our website. (Products implementing OMG specifications are available from individual suppliers.) Copies of specifications, available in PostScript and PDF format, may be obtained from the Specifications Catalog cited above or by contacting the Object Management Group, Inc. at:

OMG Headquarters
140 Kendrick Street
Building A, Suite 300
Needham, MA 02494
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
Email: *pubs@omg.org*

Certain OMG specifications are also available as ISO standards. Please consult *http://www.iso.org*

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Times/Times New Roman - 10 pt.: Standard body text

Helvetica/Arial - 10 pt. Bold: OMG Interface Definition Language (OMG IDL) and syntax elements.

Courier - 10 pt. Bold: Programming language elements.

Helvetica/Arial - 10 pt: Exceptions

NOTE: Terms that appear in italics are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

0. Submission-Specific Material

0.1 Submission Preface

The EDM Council, on behalf of its members and other industry participants, is pleased to present a standard set of terms and definitions for financial industry concepts (future, separate documents), and a set of foundational modelling parameters (this document).

Chapter 0 of this document contains information specific to the OMG submission process and is not part of the proposed specification. The proposed specification starts with Clause 1 “Scope”. All clauses are normative unless explicitly marked as informative. The section numbering scheme, starting with Clause 1, represents the final numbering scheme and will remain stable throughout the submission process.

0.2 Copyright Waiver

The entity listed above: (i) grants to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version, and (ii) grants to each member of the OMG a nonexclusive, royalty-free, paid up, worldwide license to make up to fifty (50) copies of this document for internal review purposes only and not for distribution, and (iii) has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used any OMG specification that may be based hereon or having conformed any computer software to such specification.

0.3 Submission Team

The FIBO RFCs are being submitted by the EDM Council, a membership organization in the financial sector, on behalf of its members. There is therefore not a consortium or FIBO-specific submission team; instead all submissions are by the EDM Council as representative of the community of its members.

Contact:

Mike Bennett, Head of Semantics and Standards

EDM Council Inc.,

10101 East Bexhill Drive, Kensington, MD, USA

mbennett@edmcouncil.org

0.4 General Requirements

The FIBO initiative started out as a collaborative project within the Enterprise Data Management Council, with the stated aims of:

(i) Defining common terms, definitions and business relationships (i.e. common semantics) for the financial services industry, and

(ii) Presenting this for review, validation, completion and sign-off by industry subject matter experts (i.e. presentation)

The two business requirements for common semantics and for visual and textual presentation of these to industry subject matter experts led to the creation of the “Semantics Repository”, with the additional strong mandate to “keep the philosophy out of sight”, meaning that the repository was built along semantic web principles but with the more technical views of semantic web notations kept out of sight of industry subject matter experts.

This initial Semantics Repository was built using an early version of the Object Management Group’s standard Ontology

Definition Metamodel (ODM) which at the time was in an early draft. Certain features of the then draft of ODM were not amenable to the above-described commitment to present the subject matter to subject matter experts without the intrusion of some level of technical modeling language constructs, and so considerable modification and customization of that ODM draft was undertaken. The resultant model, which was maintained within the Sparx Enterprise Architect modeling tool, was displayed on a custom-built website in the form of tables and diagrams at varying levels of detail and complexity, but free of semantic web notation.

This project brings the content developed within the above modeling framework and refactors the model within the latest version of the ODM standard. Many of the customizations which the EDM Council undertook for the reasons described above now have parallels in the most recent versions of ODM (versions 1.0 and the upcoming version 1.1) and so it was deemed possible to retain the commitments made to business consumers of the content while upgrading the model to a fully conformant rendition of ODM.

0.4.1 EDM Council Involvement with the OMG

The EDM Council is submitting the Semantics Repository as a series of specifications under the FIBO umbrella for the following reasons:

- To leverage the OMG to manage these standards within a well-founded process as provided by the OMG;
- To bring our application of the OMG's Ontology Definition Metamodel (ODM) standard up to date, based on our earlier usage and adaptation of what was an early draft of that specification.

0.5 Future Changes to this Specification

It is anticipated that aspects of this specification may need to be updated on an ongoing basis, while others may not:

- Architecture: this is intended to remain relatively static. Updates to this part of the specification shall follow the same principles as normally apply to OMG specifications for modeling languages;
- Content: the content in this specification is considered foundational to the remaining FIBO specifications and as with the content in those specifications it is expected that this will need to be extended and refined on an ongoing basis;
- Conformance: the conformance points described in this specification shall follow the same principles as normally apply to OMG specifications for modeling languages, but it is anticipated that additional conformance points may be added to the ones in this specification on a more regular basis as new ways of applying the content of the remaining FIBO specifications are identified, for example in the creation of operational ontologies which may be determined to introduce new ways of applying this content in a way which is determined should be defined as conformant.

0.5.1 What is "Content"?

For the purposes of this and other FIBO specifications, "Content" is defined in Section 4 of this document as "Subject matter or meta-content", while Subject matter" is defined as "Information about things in the universe of discourse; the essential facts, data, or ideas that constitute the basis of spoken, written, or artistic expression or representation; often : the substance as distinguished from the form especially of an artistic or literary production."

All content in the FIBO specifications is subject matter in the form of ontologies, that is models in which the model content has as its referent some feature of the business world or problem domain. This is described in further detail in the Conformance section of this specification, under "Model Theoretic Conformance".

0.5.2 Content Change Management

This specification anticipates some refinement in the OMG's processes to provide a rigorous treatment of content specifications, such that that content may be updated on a more regular basis than would be expected for specifications of modeling languages. Whereas a modeling language should remain stable so that people may create content using that language, a content specification of necessity contains material which is itself about some subject matter (in the case of an ontology, about some real world problem domain), and this content is likely to need to change on a more regular basis. The formal arrangements for changes in content generally are well established in the software engineering community, and are as rigorous as those for modeling languages changes, but of necessity operate on a faster time scale.

In the case of the FIBO specifications, it is expected that updates to content will need to be made on a regular basis either every three months or every six months, following publication of the initial versions of these specifications. This is to account for the rapidity of change in the subject matter which is modeled in these specifications: new instruments are invented by financial firms, new regulatory requirements are laid down by lawmakers, new risks identified in the marketplace and so on, and these must be reflected in the appropriate FIBO specifications as soon as this can be done in a controlled basis and in line with the rigorous processes set out by the OMG..

1 Scope

This specification is a model of business concepts that are represented by finance industry terms as used in official regulatory and financial organization documents on the subject of Business Entities. By ‘concept’ we mean the meaning of a concept, rather than any given term that represents it.

1.1 Executive Summary

This specification describes the Financial Industry Business Ontology for Business Entities. This is a model of business concepts, as described in this section, and is configured to as to be able to be presented to industry subject matter experts in such a way that those domain experts are able to review and validate the business content without any formal technical training requirement.

The FIBO for Business Entities specification covers two broad aspects: the content of the model as a set of business concepts, and the presentation of this content for business domain expert review. The latter requirement is important both for the use of the content as a formal business conceptual model within users' technical development activities, and for extension of this model content (either locally by potential users or for the submission of future model content for this specification).

This specification describes the nature of FIBO for Business Entities, the disposition of different aspects of the standard, and the detailed modeling notation which has been employed. A number of informative annexes are provided which are intended to assist potential users with the adoption and implementation of this specification.

1.2 Scope of Financial Industry Business Ontologies

1.2.1 What FIBO Is

The content that comprises the Financial Industry Business Ontology (FIBO) is documentation, interpretable in formal logic, of the concepts represented by finance industry terms as used in official financial organization documents such as contracts, product/service specifications and governance and regulatory compliance documents.

FIBO concepts are documented using two forms of definition:

1. a structured specification of the concept as a set of qualifiers of the concept ‘thing’ *(anything perceivable or conceivable)* specified as formal axioms.
2. natural language definitions which represent the structured specification in natural language with wording typically used in the finance industry.

Thus FIBO is a formal, meaning-centric dictionary and model for the special-purpose language *(jargon)* used in the finance industry. This may also be referred to, in some development process terminologies, as a ‘Business Conceptual Model’ and that is a term used widely within these specifications to describe the nature and intended usage of FIBO specifications.

1.2.2 FIBO and MOF Metamodeling Concepts

As with all kinds of dictionaries, FIBO is simply content: i.e. information or data about financial business concepts and the terms used to express them. In OMG MOF framework terms, FIBO is an M1 model, just like any other business document, web page or data content. Note that in the case of FIBO, the levels of abstraction represented by M1 and M0 refer not to the abstraction of data models but of real things - for example a class or set of real things (M1) would include the concept of a bank, while an individual such as Barclays Bank plc. is an M0 individual. That is, the answers to the questions "What is the level of abstraction?" and "What is this a model of?" do not depend on one another.

The FIBO content is interchanged as such using M1 XML content documents that:

- Either use an XSD that is generated from the ODM MOF/XMI metamodel as extended by FIBO,
- Or are MOF instance models of the ODM MOF/XMI metamodel as extended by FIBO.

FIBO content may also be interchanged using the OWL notation directly, as RDF/XMI OWL ontology files.

A dictionary is not a metamodel. Dictionaries have no metamodel levels. All terms in a dictionary including the terms that define the dictionary content itself are at the same level. Dictionaries are easily and naturally extendable, as happens all the time in the culture. The same is true for FIBO.

FIBO can be further distinguished from metamodels or document/message/data/reasoning schemas of all kinds.

- FIBO models things in the real or planned world of the finance industry. Instances of the concepts in FIBO are always those real or planned things.
- FIBO will not contain instances of its own concepts. FIBO contains only concepts - even if those concepts have just single instances in the real or planned world of finance.
    - Exceptions are made in three instances:
        - Instances which are needed in order to define property which refer to them;
        - Classes of thing which are defined extensionally; and
        - Examples
- FIBO is not any kind of a data, message or reasoning metamodel, although it adds great value to both. It does not model document/message/data content or schemas optimized for reasoning.
- FIBO will not include concepts about the structure of content, messages, information or data, even if that data is in turn about the finance industry.

1.2.3 Applications or Uses of FIBO

One of the key benefits of FIBO with respect to data, message or reasoning metamodels is that it can provide a semantic anchor firmly rooted in the concepts as understood and used by people in the finance industry for each of their components, and the terms used for them. FIBO allows one to create logical models with reference to its formal semantics so that those logical models inherit their semantics from FIBO.

FIBO allows disambiguation of new and existing regulation. To the extent that regulatory requirements are referenced to the formal semantics in FIBO, terms referred to in these regulatory requirements, or in reports that are mandated, would be semantically unambiguous.

One important purpose (for many businesses) is that the formal business definitions are used in legal documents such as contracts, terms and conditions of sales and payment, IP protection, compliance reports, and to underpin less formal language used in advertising and customer-facing websites. These language resources would typically be created and maintained as part of the knowledge management programs of organizations that apply FIBO to their business communications needs.

The business terms and definitions in this specification may be used as a reference model to which firms would tie their own proprietary models (semantic models); and also as a catalog for all of the relevant data models.

1.2.4 FIBO's Distinguishing Features

The FIBO model was built both to be business model of concepts, and to be used as such within any technical development lifecycle. That is, the FIBO model was built not only to represent business concepts but to present these representations to business domain experts. The principal distinguishing features of FIBO are therefore:

1. it is a model of business concepts as described in Section 1.2.1 and

2. it is a way of presenting this content to business audiences.

1.2.5 How FIBO is Different from Data Models

When comparing different kinds of model, the FIBO model, as outlined in Section 1.2.1, is the type of model which is referred in model terminology to as a "Business Conceptual Model".

The distinctions between the scope of the FIBO model, and that of both logical and physical models, is further described in Annex B. In summary:

- Items in the FIBO model represent entities in the 'domain of discourse', in this case business entities

- Items in a Logical Data Model represent data constructs which comprise information about those entities
- Logical data models are typically designed for efficiency and reuse of constructs without reference to the semantics of the data elements – so for example data elements may be re-used in different contexts to represent different meaningful concepts
- Physical models represent the deployment of some logical model design in some specific physical architecture

1.2.6 FIBO as a Terminological Ontology

The model described in this specification and produced and maintained according to the principles set out in this specification is a Terminological Ontology. That is, it contains not only formalization of concepts (an ontology) but also contains formal written definitions for each construct.

The meanings of terms contained within the ontology described and included in this specification are therefore formalized in two separate and complementary ways:

- Via the formal axioms stated using the ontology notation (OWL) and

- As human readable definitions.

Note that the human readable definitions have been constructed by and with the input of business subject matter experts. These are not intended to be formally structured definitions in the sense defined for example in the SBVR standard, but rather are written definitions of the meaning of the concept as the practitioners in the industry themselves see that concept.

Many definitions have been derived from definitions of similar terms, or data elements corresponding to those terms in industry data standards or industry messaging standards. These have been adapted where necessary to ensure that they are descriptive of the thing or fact itself and not descriptive of data elements conveying data about those things or facts, and have then been reviewed by industry subject matter experts to ensure that such adaptation accurately captures the sense of the business concept. In other cases (for example where the definition in a data or message standard was incomplete, too context-specific or was tautologous), a fresh definition has been framed by or with the help of industry subject matter experts.

1.2.7 Relation to Ontologies in Semantic Technology Applications

An ontology, regardless of how it is to be used, sets out formally a representation of items in a real-world domain of discourse. There are two distinct uses to which this applies:

- A Business Conceptual Model as described in this specification – this uses the full expressive power of the chosen notation to formally define items in the domain of discourse, without reference to any application constraints (because it is not an application)
- An Operational Ontology is constrained to operate within the parameters of some specific semantic technology. Typically, this will contain a sub-set of the constructs in the business conceptual ontology, such that that sub-set represents a decidable ontology.

It is necessarily the case that when something is to be used in an application, there will be technical constraints imposed upon that application. This is true when the application uses an ontology, just as it is true when the application is designed using other technologies.

The technical constraints which necessarily apply to an operational ontology necessarily do not apply to the modeling of the business domain for a Business Conceptual Ontology.

That is, the existence of some technical constraint in the application domain should not in any way influence the way in which business facts are formally captured and modeled in a business conceptual ontology. Rather, the formal requirements which apply to any deliverable which is a business conceptual model are to be applied to this model.

2 Conformance

Audience: Technical, semantic technology and standards audiences.

This specification does not describe a modeling language for which there would potentially be conformant and non conformant implementations. Therefore the kinds of conformance measures and statements which would be present in a technical specification for a modeling language standard do not apply to FIBO.

2.1 Overview

Conformance points are defined for the following types of usage of FIBO:

- Conformant extensions of FIBO within the FIBO modeling paradigm
- Conformant representation of FIBO content for business domain consumption
    - Diagrams
    - Spreadsheets or tables
- Conformant technical applications of FIBO content in:
    - Operational ontologies

Note that in addition to conformant applications, there are a number of scenarios in which someone may make use of the FIBO ontologies as a business conceptual model while applying their own design to meet their requirements. It is not possible to define conformance points for each of the possible ways in which one may legitimately develop a conventional database application or an operational OWL ontology that would be a good application. The non normative annex [Annex G] describes a number of acceptable model architectures which may adequately reflect the material in FIBO Foundations and any of the other FIBO specifications.

Any one of the following may claim to be conformant with one or more specification, module / section or ontology, and/or one or more aspect of FIBO:

- Local extensions for use as business conceptual semantic models locally
- Representations of FIBO, parts thereof or extensions thereof

For any of the above to claim conformance to FIBO, they must first identify which of the following conformance points they are asserting conformance to.

Conformance points are summarized in the table below:

Conformance Point	**Applications**	**Styling**
FIBO Full Conformance	FIBO Extensions in ODM	FIBO-Full Conformant extension for [specification] or [new]
	FIBO Extensions in OWL	FIBO-Full Conformant extension for [specification] or [new]
	Operational Ontology	FIBO-Full conformant operational ontology for [specification]
	Other conceptual formats	FIBO-Full conformant representation of [specification] in [format]
	Logical data models	FIBO-Full conformant data model for [specification]
FIBO Conformance without Archetypes	FIBO Extensions in OWL	FIBO Non-Archetyped Conformant extension for [specification] or [new]
Ontology conformance	Operational Ontology	FIBO-Full conformant operational ontology for [ontology]

Conformance Point	Applications	Styling
Presentation conformance (diagrams)	Model diagrams	FIBO-conformant diagram of [subject matter]
	Tools which are able to create diagrams from FIBO content	FIBO-conformant editor
Presentation conformance (tabular)	Spreadsheets	FIBO-conformant table or spreadsheet for [ontology / module]
	Tools which are able to create reports or spreadsheets from FIBO content	FIBO-conformant editor

2.1.1 Basic Conformance Points

The following basic conformance points are principally intended to be included in the FIBO-Full conformance point and others:

Conformance Point	Description	Section
FIBO Model Theoretic Conformance	Conformance with the model-theoretic principles set out for all FIBO and FIBO-derived ontologies	2.2.1
FIBO-Conformant ODM Application	Conformant use of the sub-set of ODM defined in the Architecture section of this specification	2.2.2
FIBO Archetypes Conformance	Conformance with the use of archetypes as set out in this specification.	2.2.3

2.1.2 Conformant extensions of FIBO

Based on the above conformance points, the following conformance points may be asserted for extensions to FIBO itself:

Conformance Point	Description	Section
FIBO-Full Extension	Fully conformant extension to FIBO in the FIBO modeling ecosystem, including full rendition of standard metadata, archetypes and so on. Must include FIBO Model Theoretic conformance and FIBO-conformant ODM application.	2.3
FIBO Non-Archetype Extension	Conformant extension to FIBO in OWL modeling tools, without the inclusion of the Archetypes constructs and related metadata.	Not 2.2.3

FIBO Extensions conformance is the same whether conformance is asserted for a local extension of FIBO within a firm for the uses described for FIBO, or for the submission of proposed new content for future iterations of the relevant FIBO specification or a new FIBO specification.

Both of these conformance points must include FIBO model theoretic conformance. FIBO-Full extensions (i.e. semantics models within the FIBO ODM-based modeling paradigm) must also include conformance with the FIBO application of ODM and the use of FIBO archetypes.

Note that in the FIBO ODM ecosystem it is expected that any extensions to FIBO will include archetypes and where appropriate may add or propose additional archetypes of their own which are conformant with the principles set out here. Archetypes may also be included in OWL modeling environments, but these are of less immediate relevance and may be harder to view and validate. For this reason, a separate conformance point is defined for OWL extensions to the FIBO content which do not include the Archetype annotations but are complete in every other way.

This second conformance point may also be used for proposing new content for future versions of FIBO; as long as all other aspects of conformance are adhered to, it will be possible for the archetypes to be added in retrospectively to render such proposed content as FIBO-Full conformant content for future versions of a given specification or for future new specifications.

2.1.3 Conformant representation of FIBO content

There are two conformance points for this:

Conformance Point	Description	Section
Diagram presentation	Conformant presentation of the model content in diagrammatic format	2.4.1
Tabular presentation	Conformant presentation of the model content in tables or spreadsheets.	2.4.2

This means that it is possible to assert that a spreadsheet is conformant to FIBO, via the Tabular Presentation conformance point.

Any tool which asserts support for one or other or both of the above conformance points must be able to import the available FIBO content in one or other of the available serialization formats (XMI or OWL), and produce diagrams and/or tables as asserted, which conform with the requirements defined for this conformance point.

At the time of writing, most if not all OWL editing tools are not able to present their content in accordance with the above conformance points.

2.1.4 Conformant technical applications of FIBO content

2.1.4.1 Operational ontologies

Conformance Point	Description	Section
Specification Full FIBO Conformance	Conformance with all the ontologies in a given FIBO specification along with conformant use of the terms in the FIBO Foundations specification, including the upper ontology Lattice. Must also include FIBO Model Theoretic conformance.	2.5.1.1
Ontology conformance	Conformance with a specific ontology within a module of FIBO along with all the related FIBO content in the non-included modules in the same FIBO specification, in other FIBO content specifications where these are formally imported by the ontology for which such conformance is claimed, and in the FIBO Foundations specification. Must also include FIBO Model Theoretic conformance.	2.5.1.5

2.2 Basic Conformance Points

2.2.1 Model Theoretic Conformance

2.2.1.1 Relationship to Subject Matter

Each model element which is a class, an object property or a datatype property shall correspond to some item in the real world. No model element shall refer to some technical construct such as a database field, internal identifier, database key and the like. Each model element is a linguistic object which the semantic language is based on. The elements are considered as statements and the truth of the statements is defined by membership in the underlying relations. A semantical system is defined as any map whose domain is a non-empty set of model elements and whose range is included in {true, false}. That is, the individual model elements are mapped a semantic truth function where the mapping is true when a correspondence to a real world semantic item exists and false otherwise..

2.2.1.2 Information Constructs

An exception to the above requirement is made for information constructs which are themselves an important and publicly shared part of the business domain, such as publicly issued identifiers. These are styled by the archetype of "Information" or some sub-type thereof. Reference may only be made to information constructs which are not part of some system design but which are shared across the industry, such as security identifiers, ratings codes and the like. In each such case, there shall be some formally identified scheme in which the code in question is defined. Since these items are also linguistic constructs, they conform to our formal concept of mapping to the range {true, false}.

A suitable test for types of "Information" which are considered real for the purposes of this application of the model relationships is whether that information is publicly shared or, if private, made available across the business supply chain. Examples include securities prospectuses, published indices, interest rates and so on. If the information does not have a "Publisher" (which shall be indicated in the model by a suitable relationship to some entity or party which may be the publisher) then the modeler should question whether the information construct represents some real thing. In the absence of a formal "Publisher" or other semantically represented fact about the provenance of the information construct, the decision to include it should be ratified by other competent business domain experts as part of the review and update process. An alternative way of considering this idea of publication is to ask whether the truth function is resolvable for the particular item, that is, would the “public” be able to ascertain whether the model element was in fact “true” or “false” without ambiguity. If not, then the element should most likely *not* be included.

2.2.2 FIBO Conformant ODM Application

A model extension is conformant with this requirement if it uses the Ontology Definition Metamodel (ODM) standard as described in the Architecture section of this specification (section 8). In particular, only the ODM base classes defined in Section 8 may be used.

2.2.3 FIBO Archetypes Conformance

A separate conformance level is described for the use of the archetypes for classes of Thing and for relationship facts (OWL Object Properties). The archetypes provided as part of FIBO are intended to provide a formal foundation for how new classes of each archetype are to be used. For this conformance point:

- All classes shall have an archetype.
- Classes of a given archetype shall have as an ancestor the class which defines that archetype.
- The necessary, defining properties of the archetype are modeled as properties of the class which defines the archetype (the archetypal class). Classes which are of that archetype may extend, add to or specialize those properties.
- The relationship facts of the archetypal class each have relationship archetypes. The net result is a set of necessary facts about each archetype, which is to be respected for each class which is defined as being of that archetype.
- All relationship facts shall have an archetype.
- The properties of a class archetype, where these extend or specialize the relationship facts about the archetype, shall have the same relationship fact archetype as the corresponding relationship fact which it extends or specializes.
- Simple facts do not have an archetype.

2.3 Conformant Extension of FIBO Content

Audience: Semantic Modelers

These conformance points apply both to extension of the model content for use locally (conformant application of this specification), and for the preparation for submission of new model content for future versions.

Extensions to the FIBO content must also comply with the following conformance points:

- Model Theoretic Conformance
- Semantic Conformance
- Conformant use of Owl as specified in the relevant W3C specifications

That is, ontological conformance breaks down into model theoretic conformance, syntactic conformance to the OWL modeling language, semantic conformance (whether the terms are meaningful) and (if asserted), appropriate use of the archetype concept in FIBO.

This is the same whether conformance is asserted for a local extension of FIBO within a firm for the uses described for FIBO, or for the submission of proposed new content for future iterations of the relevant FIBO specification or a new FIBO specification.

2.3.1 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

2.3.1.1 Conformant Taxonomy (Classification)

There are two ways of stating "is a" about something - via an object property and via generalization hierarchies. The generalization "is a" relationship (sub class relationship in RDF/OWL) represents and only represents a relationship in which the entity at the "bottom" end of the relationship is genuinely a kind of the thing at the top of the arrow - that is, that it inherits all the facts about that class of thing. Other styles of "being" something should be represented with object properties (relationship facts).

In many cases, the correct use of an "is a" relationship is via the model partitions for independent versus relative entities. These are described in the section below on conformant application of partitioning (Section 2.1.4.3).

2.3.1.2 Conformant application of Partitioning

The partitions are:

- First-, second- and third-order constructs
- Concrete versus abstract partitions
- Continuant and occurrent partitions

2.3.1.2.1 First-, second- and third-order constructs

First-order: Independent Things

A concept shall only be a sub-class of some first-order construct (that is, a concept which has the class "Independent Thing" as an ancestor) if the definition of that concept holds across multiple contextual uses. That is, the concept should retain the same meaning regardless of context.

Example: A Legal Entity.

Second Order: Relative Things

A concept shall only be a sub-class of some second-order construct (that is, a concept which has the class "Relative Thing" as an ancestor) if the definition of that concept has a definition which is wholly dependent on the context in which it is used.

Example: A Security Issuer.

Relative Things shall have an identity of "identity" which has a range which is the thing which fulfils that role. This is usually an Independent Thing, but may also be some other Relative Thing.

Relative Things should also have a relationship to the context in which they are defined, of the form "defined in the context of". This should normally have as its range some Third order or Conceptual Thing. For certain relative things which are aspects of some independent thing (descended from the concept "Aspect"), the range of this relationship shall be that thing of which it is an aspect.

Relative things are normally specializations of existing archetypal relative things, such as Party, Actor or Underlying. New Relative Things shall be disposed within the relevant taxonomic hierarchy of relative things where this exists.

Relative things include "Part" concepts. A concept shall only be a sub-class of some "Part" (that is, have "Part" as an ancestor) if the concept as defined means and only means that it is defined in its role as that part of that thing. Things which may go on to become parts of something, or which are things in their own right but which normally fulfill the role of parts, shall be modeled as Independent Things, with their role as a part defined separately as a Relative Thing.

Example:

A wheel is an Independent Thing;

The Nearside Front Wheel is a Relative Thing.

Third Order: Mediating Things

The third order represents contexts in which relative things are defined. These approximate to business or other contexts.

A concept shall only be a sub-class of some third order construct (that is, a concept which has the class "Mediating Thing" as an ancestor) if it is properly defined only as a kind of context in which the definitions of relative things hold, and without which those relative things would have no meaning.

Use of this Partition

Concepts shall not have ancestors from more than one of these partitions, since these are disjoint. If a concept appears to be properly a sub-class of concepts that belong in more than one of these partitions, then one of those classes or one of its ancestors has been improperly allocated to its partition - for example, something which was thought to be a context is really an independent thing with a similar name to some concept.

2.3.1.2.2 Concrete versus abstract partitions

A concept shall only be a sub-class of some concrete thing (that is, a concept which has the class "Concrete" as an ancestor) if it represents some concrete entity in the business domain. Concrete entities for the purposes of this specification include information constructs whether these are dematerialized or not, so for example Share is regarded as a Concrete Thing whether or not it exists in paper.

Example: Swap Contract; Limited Partnership.

“Abstract” defines things which are abstract by nature, for example strategies or goals. The definition for these is “A concept or idea not associated with any specific instance.” A concept shall only be a sub-class of some Abstract thing (a concept which has the class "Abstract" as an ancestor) if its meaning and definition represent something which is not capable of implementation as some concrete instance.

Example: Portfolio Strategy.

2.3.1.2.3 Continuant and occurrent partitions

A concept shall only be a sub-class of some continuant construct (that is, a concept which has the class "Continuant Thing" as an ancestor) if it represents some entity which has some ongoing existence over a period of time.

Example: Swap Contract; Limited Partnership.

A concept shall only be a sub-class of some occurrent construct (that is, a concept which has the class "Occurrent Thing" as an ancestor) if it represents some entity which has no ongoing existence over a period of time. That is, classes in this partition shall only represent concepts which have their proper definition framed in terms of some occurrence.

Example: Payment Event,

Events which continue over some period of time but are properly framed with reference to their time component, for example business processes and process activities, shall be defined as Occurrent Things.

2.3.2 Labeling

Labeling shall be visible for all model constructs which are to made visible to business subject matter experts and which are to be reviewed and validated to them. These labels have the following formal requirements, in order to be conformant with their intended usage, and in respect of the presentation rule that no notation shall be presented to business domain reviewers which either is or which appears to be in some technical notation.

- Labels shall not be in camel case
- Labels shall represent a plain English name (in US English spelling), which is the label chosen by the business domain experts themselves as being that which is most usefully attached to the meaning of the term being labeled.
- Labels shall not contain long chains of qualifiers which make them diverge from normal business English naming of the concepts
- Labels do not need to be unique across the model
- Labels shall not be in the form of, or contain, acronyms (including business acronyms) except where these are the only term by which the term may be referred in the business domain (for example "CDO Squared").

2.3.2.1 Class Labeling

In addition to the above requirements:

- Labels for each class of "Thing" shall be unique within each ontology (the model consists of several discrete ontologies).

2.3.2.2 Object Property Labeling

Labels for each relationship fact (OWL Object Property) shall have a formal name of the form "Domain Class predicate Range Class", respecting the casing convention whereby classes of "Thing" are named in Capital Case and relationship predicates are named in lower case.

Relationship Facts shall additionally have a "short" label which takes the form of the predicate only. If the tool supports separate labeling for the association component of UML Association Classes, this shall be populated with the "operationalLabel" annotation metadata content for that object property. Where this cannot be achieved within a particular notation then separate labels shall be applied using the content of the "operationalLabel" annotation metadata.

Predicate names shall not be required to be unique within each ontology

Long labels (full names) of relationship facts shall be unique within each ontology.

2.3.3 Model Consistency

Certain aspects of conformance may be tested for or verified automatically. These include checks for consistent application of the model constructs.

There are several definitions for ‘consistency’ in the ontology literature. The consistency check which can be applied to this conceptual ontology is known as "Description Logic" checking, that is, the ontology may be checked to ensure that it is description logic complete.

Consistency checks will pick up, among other things, inconsistent application of the mid level ontology and upper ontology constructs. Failures in such checks indicates the possibility that some such has been misapplied.

There are constraints which apply to all ontologies, whether they are full business conceptual ontologies or operational ontologies. For example they should always be internally consistent (there should be no contradictions). There should be Consistency checks applied which are appropriate to the form of specification of the ontology.

No derived statement within the ontology shall be a negation of any other statement within the ontology.

2.4 Conformant Presentation of Model Content

It is a requirement of this specification that content of the models is made available to people in the business domain in one or more of a set of diagrams and tables which are described in this specification.

The context of this section is the use of FIBO within UML modeling tools. The FIBO content from the repository is ingested into the UML tool - see Figure 1, section 8 (architecture).

The FIBO presentation notation was developed in order to fit into formal quality assurance processes, in which it is a requirement for a business conceptual model that it be reviewed and validated by business subject matter experts. This remains a requirement for new material which is to be presented as potential updates to the FIBO content in future iterations. It is also anticipated that users of the content locally, whether for conventional model driven development, for semantic application development or for the integration and mapping of disparate systems, databases and data feeds, should also need to position this material within their own quality assurance processes in precisely the same manner, and it is for this reason that the presentation aspects of FIBO were developed and should be adhered to in any manifestations of this content in UML tooling.

An implementation of this model content is not conformant is the only means for the reader to view the terms, definitions and relationships is one which requires some formal understanding of some model language such as UML or OWL in order to understand it. That is, for the avoidance of doubt, some format which contains symbols, whether diagrammatic or textual, which have a meaning other than the meaning that a reasonably educated but non-technical person would ascribe to those items on seeing them. That is, notations which require some learning of the language in question in order to understand them. The exception to this is the few symbols which are explained in this specification.

An implementation of this model content within one of the editing environments described elsewhere in this specification is conformant with this specification if the diagrams made available to the business domain are understandable without recourse to a knowledge of modeling languages other than the explanations given in the annexes to this specification.

There are two conformance points for this:

Conformance Point	Description
Diagram presentation	Conformant presentation of the model content in diagrammatic format
Tabular presentation	Conformant presentation of the model content in tables or spreadsheets.

2.4.1 Diagram Conformance

No explicitly UML notation should be present on any diagram, which does not represent some feature of OWL. An exception is made for the use of UML Boundaries, where these are used to organize and better present diagram content.

Generalization relationships shall be laid out with the "arrowhead" pointing vertically upwards, in either the vertical tree style or direct style of routing.

For general guidance on diagrams creation and presentation please refer to the non normative Annex F on Extending the Model Content.

2.4.2 Tabular Reports Conformance

Tabular reports are specific in two flavors, both of which have the conformance requirements described in this section in order to meet the stated requirement that they may be reviewed and validated by business subject matter experts. These may be rendered as spreadsheets or as textual documents in a tabular layout.

The two flavors are:

1. Basic Table
2. Extended Table

2.4.2.1 Basic Table

The "Basic" tabular format shall show only the following entries:

- Term
- Definition
- Synonym

These shall be labeled as such.

This table shall only show those constructs from the model content which represent meaningful business concepts, and not the additional constructs which deal with the set theoretical logic of the model. That is, the basic table shall show only:

- Class
- Relationship Fact
- Simple Fact
- Union Class

2.4.2.2 Extended Table

Extended Tabular reports shall be presented which conform with the following requirements:

The extended table shall have column entries for each of the basic model features, as follows:

- Term
- Definition
- Synonym
- Range of simple facts (titled as "Simple Type")
- Range of relationship facts (titled as "Related Thing")
- Multiplicity (labeled as "multiples")
- Additional metadata may or may not be shown, at the discretion of the modeler and as appropriate to the intended usage, for example review notes annotations.

The model constructs which are not shown in this tabular format are only those which comprise relationships among relationships, namely sub-property relations and inverse relations.

The following model constructs shall be included in the Extended Table reports, in or near the following order:

- Class
- Parent
- Union Relationships
  - labeled "In Union" when reported for members of the union
  - labeled "Union Of" when reported as the relationships from the Union Class
- Relationship Fact
- Simple Fact

- Union Class
- Disjoints (labeled "mutually exclusive")
- Individuals
- 'typeOf ' relationships from Individual to Class (labeled "type of")

Relationships shall only be included once in all reports across the model, and this shall be for the class which is the domain of that property. The exception to this is the logical union relationship owlUnion (represented using a UML covering GeneralizationSet construct); this shall be reported from both ends but with separate meaningful labels for each end as shown.

The intention of these requirements is that the report does not resemble auto-generated reports from technical designs, but shows each type of fact, once only and in a logical order.

2.5.1.5 Ontology conformance

Description: Conformance with a specific ontology or a named set of individual ontologies within a module of FIBO along with all the imported FIBO content in the non-included modules in the same FIBO specification, in other FIBO content specifications where applicable, and in the FIBO Foundations specification.

Not all content need be included from the rest of the module, the rest of the specification or the FIBO Foundations specification; however each class in the operational ontology must have a generalization hierarchy leading all the way up to OWL Thing which reflects that of the complete FIBO content.

2.5.1.6 Annotation Metadata

For each of the conformance points described in this section, a conformant semantic representation of the FIBO content may or may not use as many of the annotation properties as are considered necessary for that application.

Annotations are rendered as OWL Annotation Properties. These may not be restyled or re-framed as object properties in a conformant application. That is, an application may do this but may not assert FIBO conformance.

Under this conformance point, the ranges of annotations in an semantic representation of the FIBO content may be those given in the corresponding FIBO conceptual ontologies, or they may be replaced or simplified, for example by replacing a range which is a class (such as a standards body, a document or a website) ,with a literal string. This remains within the scope of this conformance point.

3 References

3.1 Normative References

The following normative documents contain provisions which, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply.

- Web Ontology Language www.w3c.org/owl Version 2
- RDF Schema at the version referenced in ODM version 1.1.
- Ontology Definition Metamodel (ODM) version 1.1 [OMG specification number to follow]
- OMG recommendations for the application of DC and SKOS metadata - finance/2011-12-02
- OMG AB recommendations for ontology metadata – AB/2013-02-02
- XMI: formal/2011-08-09
- Dublin Core (DC) at: http://dublincore.org/
- Simple Knowledge Organization System (SKOS) http://www.w3.org/TR/2009/REC-skos-reference-20090818
- ISO 1087-1:2000 Terminology — Vocabulary — Part 1: Theory and application
- W3C Organization Ontology at: http://www.w3.org/TR/vocab-org/

3.2 Non Normative References

The following informative documents are referenced in this specification or in parts of the Annexes:

- "The Axiomatic Method in Business Economics: A First Approach", Pellicelli, G., Abacus (December 1969), pp. 119-131.
- "Introduction to Model Theory and to the Metamathematics of Algebra", Robinson, A.,. Amsterdam: North-Holland, 1963.
- "Information Technology - Common Logic ISO/IEC 24707:2007" http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39175
- "Logic and Structure", Van Dalen, D., second ed. Berlin: Springer-Verlag, 1983.
- "A Mathematical Introduction to Logic", Enderton, H. B., Orlando: Academic Press, 1972.
- "Mathematical Logic: An Introduction to Model Theory", Lightstone, A. H., New York: Plenum Press, 1978, edited by H. B. Enderton.
- "Ontology Metadata Vocabulary" (OMV) - http://omv2.sourceforge.net/
- "Towards a General and Axiomatic Foundation of Accountancy", Mattessich, R, Accounting Research (October 1957), pp. 328-355.
- "Zachman Framework", Zachman, J., at http://www.zachman.com/

-

3.3 Changes to Adopted OMG Specifications

This specification does not change or replace any OMG specifications.

4 Terms and Definitions

For the purposes of this specification, the following terms and definitions apply.

Content

Definition: Subject matter or meta-content.

Business conceptual model

Definition: A model which represents and only represents business subject matter without reference to the design of any solution or data model representation.

Definition:

Business publication

Definition: Representation of a subject matter view in a form that is understandable and usable by business users.

Example: Text document, web page, audio recording, interactive search dialog

Business subject matter

Definition: Subject matter that defines and describes the kinds of people (and the roles they play), organizations and other things that an enterprise has to deal with in the course of its operational business, regardless of how this content is presented to the people in the organization (e.g. in text documents, web pages, audio broadcasts).

Example: Business concepts, such as: *OTC derivative*, *business day*

Example: Relationships between business concepts, such as: *swap transaction has ISDA confirmation*

Example: Constraints, such as: Each ISDA confirmation is of exactly one swap transaction

Example: Descriptions, such as: ISDA is the largest trade organization of participants in the OTC derivatives market.

Example: Business processes (defined in terms of the business concepts), such as:

If a Disputing Party reasonably disputes the Value of any transfer of Eligible Credit Support, then the Disputing Party will notify the other party not later than the close of business on the Local Business Day following.

Note: Business subject matter is mainly about kinds of thing, but may include individuals, in three roles: (1) as one-of-a-kind things referenced in the subject matter, such as *ISDA*, *Dodd-Frank Act*, *EC Treaty*; (2) As types defined by enumeration, such as the currencies in which a trading business maintains accounts; (3) in examples.

Note: Business subject matter is usually scoped by area of business jurisdiction (or something similar), such as, say, derivatives trading. The business subject matter is about the business of derivatives trading.

Other areas of responsibility in the enterprise have different subject matter. For example, the IS department's subject matter includes information models of things in the operational business (including derivatives trading). The finance department's subject matter includes financial models of things in the operational business.

From the derivatives trading perspective (the relevant parts of) these information and financial models would be considered meta-content.

Business subject matter view

Definition: Subset of business subject matter that is intended to be presented in some business publication.

Example: Concept definitions; relationship definitions with constraints.

Extension

Definition: The membership of some class of thing. This is distinct from its intension, that is the properties intrinsic to that class of thing. In applying the intension of some class to some collection of individuals, one arrives at the extension of that class for that collection.

Extensional

Definition: Logic explicable solely in terms of extensions; ignoring differences of meaning that do not affect the extension.

Extensional Definition of Class Membership

Definition: The definition of membership of a class by direct articulation of those members (that is, by articulation of the Extension of that class.

Intension

Definition: The properties intrinsic to some class of thing.

Intensional

Definition: Logic (of a predicate) incapable of explanation solely in terms of the set of objects to which it is applicable; requiring explanation in terms of meaning or understanding.

Intensional Definition of Class Membership

Definition: The definition of membership of a class according to properties intrinsic to members of that class.

Meta-content

Definition: Information about subject matter

Example: Control information, such as: date and author of last update, external source, owner

Example: Connection of subject matter items to content outside the subject matter scope, such as data model elements that correspond to them (and point to the storage of instance data).

Model-Theortic Conformance

Definition: The manner in which some model conforms with some theory about what it is intended to model and how it is intended to model it.

Ontology

Definition: A formalization of a conceptualization. For the purposes of this specification the formalization is in OWL, using ODM as a means to render this, and the conceptualization is that of business subject matter.

Operational Ontology

Definition: An ontology which is intended for use within some application.

Subject matter

Definition: Information about things in the universe of discourse; the essential facts, data, or ideas that constitute the basis of spoken, written, or artistic expression or representation; often : the substance as distinguished from the form especially of an artistic or literary production.

Taxonomy

Definition: A set of terms which stand in some classification relation to one another.

Terminology

Definition: The overall disposition of ontologies of concepts and vocabularies of terms, in relation to one another.

Vocabulary

Definition: A set of words, each giving one or more formal definitions which apply to a meaningful concept that is referred to by that word.

Definition:

5 Symbols and Abbreviations

5.1 Symbols

There are no symbols introduced by this specification.

5.2 Abbreviations

The following abbreviations are used throughout this specification:

- OWL – Web Ontology Language
- ODM – Ontology Definition Metamodel
- RDF – Resource Definition Framework
- SME – Subject Matter Expert
- UML – Unified Modeling Language
- URI – Uniform Resource Identifier
- URL – Uniform Resource Locator
- XMI – XML Metadata Interchange
- XML – eXtensible Markup Language

Additional symbols and abbreviations that are used only in annexes to this specification are given in those annexes.

6 Additional Information

6.1 How to Read this Specification

6.1.1 Audience

This specification has the following audiences:

- The standards community
- The finance industry business community
- The regulatory community
- Technical audiences
- Semantic Modelers

6.1.1.1 Standards Community

This audience is intended to be able to follow and validate the way in which this specification sets out the arrangements for the production and maintenance of model content, and the production of business facing reports and diagrams representing parts of that content.

6.1.1.2 The Finance Industry Business Community

As noted in the section on conformance (section 2) this specification includes detailed requirements for the production of diagrams and reports which are intended for consumption by business subject matter experts. This specification also contains material addressed at this audience, this being an informative annex on “Interpreting Model Content”. This audience is not intended to read and understand the remaining parts of this specification.

6.1.1.3 The Regulatory Community

As for Finance Industry Business Community.

6.1.1.4 Technical Audiences

These include but are not limited to:

- Tooling vendors and developers
- Other content providers / enriched content providers
- Business Analysts – anyone who use the model on site, whether they are a modeler, a metadata analyst, etc.
- Technology Management

The bulk of the “Architecture” section is intended to be read and understood by these audiences and by the ‘Semantic Modelers’ audience..

6.1.1.5 Semantic Modelers

Much of the material in this specification is intended to be read and understood by semantic modelers. This includes the 'Conformance' section (Section 2), the ‘Architecture’ section (Section 8) and the non normative Annex F on implementing and extending this model and proposing new model content.

The Semantic modeler audience is not the same as the technical audience, although some individuals may possess skills in both. Sections of this specification which are written for a semantic modeling audience do not require any training in any formal technology in order to understand and act upon their contents. These sections do require a clear understanding of semantics and formal logic. It is not necessarily the case that technical readers are expected to be able to read and understand all aspects of the semantic modeling material. It should also be noted that some terms which have specific meanings in one or more technology environments, may have different (or often only subtly different) meanings to the semantic modeling audience. Where both semantics and technical audiences are intended to read a section, care has been taken to try to use all of the applicable terms and qualify words which have multiple different usages to these audiences.

6.2 Acknowledgements

The following organization submitted this specification:

- Enterprise Data Management Council

The following companies supported this specification:

- Adaptive
- Business Semantics
- Citigroup
- GoldenSource
- HSBC
- Oakland University
- Thematix
- University College Cork

In addition to the above, the following companies provided significant expertise and resources in the development of the content and architecture of this specification:

- Australia and New Zealand Banking Group
- AVOX/DTCC
- Barclays Capital
- BBH
- Bloomberg
- CIBC
- CUSIP
- Fidelity
- JP Morgan
- Michigan State University
- Model Driven Solutions
- Model Systems
- Morgan Stanley
- MPhasis
- National Australia Bank
- No Magic
- Nomos Software
- Nordea Bank
- Revelytix
- Sallie Mae
- SAP
- Semantic Arts
- State Street
- Sungard
- SWIFT
- Tahoe Blue
- Thomson Reuters
- University of British Columbia
- Wells Fargo

7 Introduction

Informative.

7.1 Audiences

Readers are encouraged to read Section 6.1 on the different intended audiences for this standard.

7.1.1 Audience for this Section

The audience for this section is anyone who wishes to understand this standard, whether from a business or technical standpoint.

7.1.2 Reading this Standard

Technical audiences (in both conventional and semantic technology) are directed at the "Architecture" section (Section 8).

Business audiences (financial industry participants, regulators and others) are directed at this Introduction and at Annex A on interpreting model content (Annex B).

The business content defined in this standard is intended to be presented both in a business-facing format and in a complete, technical format. The latter is intended for consumption by technical and standards audiences only. This specification defines the content of the standard and the ways in which it is to be presented to business readers.

7.2 Specification Overview

7.2.1 Non Technical Overview

This specification provides a model of business entities terms, definitions and relationships. The model contains no technical design content and is a representation of the business entities concepts. This specification describes the technical arrangements by which this has been brought about, the requirements to be placed upon semantic modelers who are to extend this content locally or to propose updates to the model, and the requirements by which the content of this and future extensions are to be presented to business domain participants, so that they may understand and review the model content without the need for any formal technical training.

7.2.2 Technical Overview

Audience: This sub-section is intended to be read by technical audiences.

This specification describes the architecture, the use of the ODM metamodel, the usage of the ODM profile, additional supporting metadata and content of the business entities model.

The model content is developed and maintained using the Unified Modeling Language as a modeling tool framework, but with all model content built using the formal constructs of the Web Ontology Language (OWL). This is achieved using the OMG's Ontology Definition Metamodel (ODM) specification.

The use of the ODM specification in this specification is limited to a specific sub-set of OWL constructs, and is also limited to the range of UML base classes that is allowed for each of the OWL constructs that are used.

The model content is made available as serialized ODM UML in the form of XMI files, and as OWL files using the RDF/XML syntax. The deliverables are listed in Annex A.

This specification also describes additional metadata developed to support the annotation of the model content as OWL annotation properties.

This specification also describes the use of model content which is not specific to business entities, and the disposition of these within the broader model framework.

7.3 Usage Scenarios

Audience: Technical implementers (conventional and semantic technology); technology management

The model described in this specification and included with it is intended for use as a business conceptual model. As a result of the notation chosen to represent business terms and definitions, it may also be used in semantic technology adaptations, subject to suitable alterations by semantic technology developers.

These uses envisaged for the model are as follows:

- Model driven development
    - Of database schemes
    - Of message schemas
    - Of common messaging across a business unit or organization
- Semantic Technology development
- Integration of systems and / or data feeds

In addition, the model may be extended locally by potential users to extend the scope of what is modeled, prior to using such local extensions in any of the above usage scenarios.

This specification also envisages that future iterations of the model described and included herein may be proposed by any interested party, following the same processes and principles as are described for extending the model content locally within a user's firm.

7.3.1 Model driven development

Model Driven Development refers to the top town development of technical artifacts starting with a high level, business view of the requirements (for programs) or the data semantics (for data), as described in Section 1 (Scope).

In this application, the model described in and presented as part of this specification is to be used as a business conceptual model, precisely as described in the literature for such usage. That is, the model provides a formal reference, to be maintained within the development process as such and, potentially at least, extended locally with additional concepts not included in this specification which are of relevance to the development in question.

In this scenario, the model would be ingested into a UML modeling tool, and situated within a model partition for "Conceptual Models" within a broader UML repository which would also contain partitions for logical models, deployment models and so on, determined according to the formal requirements of the development process that is used within the firm.

Further inspection of the metadata provided within this model may enable the automation or partial automation of the production of logical data models, or at least of a candidate starting point for the development of the logical data model prior to the addition of keys and other database requirements.

The model described and presented within this specification supports multiple inheritance between classes, whereas most logical data models would be developed using a single inheritance taxonomy (if this is a constraint on the logical or physical models development). This model will contain metadata which defines, for multiple inheritance taxonomies, what are the facets of information by which each taxonomy has been derived. Such information can be interrogated either manually or (at least potentially) programmatically, to extract from the model a suitable single inheritance taxonomy appropriate to the requirements of the development.

Using this model within a UML tool also allows for the formal mapping between developed (or generated) logical data model constructs and the semantics constructs to which these relate. This in turn simplifies end to end validation and verification of the developed artifacts.

The model described and presented in this specification is intended to be situated within any model driven development framework, as a conceptual model. This is the case whether the development is for databases, messages or a combination of the two.

7.3.2 Semantic Technology development

As part of this specification, model content is made available in the Web Ontology Language (OWL) format, which is the format used in semantic technology applications.

However, semantic technology developers should be aware that the physical and technical constraints which rightly apply to semantic technology applications have not been imposed and will not be imposed on this model, since its primary purpose is to serve as a conceptual model at the business level.

Similarly, it should be noted that in defining the formal meanings of terms in the business domain, most of those meanings are "grounded" with reference to legal constructs, accounting constructs and so on. This may or may not correspond to instance data in the application. Typically a semantic technology application, like any other application, will operate on actual data.

There is therefore a distinct difference between the terms defined in this model to satisfy the requirements of a business conceptual model, and the terms required or to be found in an ontology that would be used in a semantic technology application.

Semantic Technology developers will therefore need to extract from the model content, some suitable and decidable sub-set of that content.

This specification does not detail exactly how to derive decidable sub-sets of the content, such as OWL-DL. It is left to the semantic technology developer to make the necessary transformations.

Some of the metadata provided with this model may assist in this. In particular, it should be possible for the semantic technology developer, by inspection of the metadata styles as "archetypes", to identify kinds of relationship which are unlikely to refer to instance data (OWL Individuals) kinds of relationship which will. This would potentially enable the extraction of a sub-set of the model content which would be amenable to semantic technology processing. Similarly, as with the conventional technology scenario described above, it may be possible to use the metadata which identifies "classification facets", to extract simpler taxonomy structures from the model.

7.3.3 Integration of systems and / or data feeds

The simplest application of this conceptual model is to simply use the terms as a common point of reference when comparing terms within different logical or physical data models. This would be of value for example when integrating different systems.

Many systems may not have a formally stated ontology for the data elements that they use, or the database schema may be considered to be the only record of the meanings of the terms therein. Typically, whenever two or more systems need to be integrated, either as the result of a merger between firms or as part of the process of installing a new system within the firm, there is a time consuming and almost open ended "mapping" exercise in which the meanings of each of the terms in each of the databases or message schemes involved in the integration, are guessed and perhaps written down.

In reality, even when the intended meanings of the elements in each database and message scheme are known, there is not an easy one to one mapping between one system and another. This is typically the result of good design: the more the design have made use of reusable common data structures, the more efficient that design is, but correspondingly the less explicit is the semantics of the terms.

In an integration project that brings together data elements from more than two systems or data feeds, the number of mappings that need to be carried out between on system or feed and another is a geometrical function of the number of such data sources and feeds. In order to have a mapping exercise which is only arithmetically related to the number of data sources and feeds, it is necessary to have a single "hub" of terms which are able to be used as a common point of reference between each of the data models.

While this can often be achieved using a single data model, in practice the limitations on data models (such as single inheritance taxonomies in many cases, though not all) mean that no one model can be found against which all terms in all data models and feeds may be cross referenced. The model presented as part of this specification, being a semantic model, contains full definitions of the meaningful concepts which may be referred to by any of the data elements in the data sources or feeds that need to be integrated, as long as this model may be extended locally to cover areas of scope which are not part of the current specification.

To use the model according to this usage scenario, one may use the UML model (as described for model driven architecture) if this is a good fit to the environment being used, or one may use the spreadsheet reports directly. The spreadsheet reports are intended as a “business facing” deliverable from this specification, but the “full terms” sections of those reports contain all the information that is present in this model with the exception of relationships between relationships (relationship inverses; sub-property relations). Since the latter exist only in semantic models and are not likely to be found in any fo the data models in a technical integration project, these spreadsheets may be used as a mapping facility.

8 Architecture

Intended Audiences: Technologists, Semantic Technologists, Standards Implementers.

This section described the architecture of FIBO, that is the structure and components of the conceptual content.

Please also refer to the Scope section (Section 1) and the Definitions (Section 4) for detailed treatment of the terms and concepts referred to in this section.

The positioning of the model with reference to other types of Architecture is described in the Scope Section 1 and is not replicated here.

8.1 Overview

The architecture is presented in several parts:

- Disposition of the standard
- Usage and restriction of the Ontology Definition Metamodel standard
- Application and adaptation of semantic modeling techniques and notations for business presentation.

These are described in the sections which follow.

8.2 Disposition

The model is maintained within a formal metadata repository. The commitments described in Section 8.5 which are made to business domain experts to provide diagram and tabular views of the model content are met from that repository. Terms and their definitions may be accessed directly through URI reference or navigated to through the interfaces provided by that repository. In addition, model content may be exported from that repository in order for users of the standard to be able to extend this locally either within conventional model driven development frameworks, or as semantic technology applications using the OWL language. Figure 1 shows an overview of these arrangements.

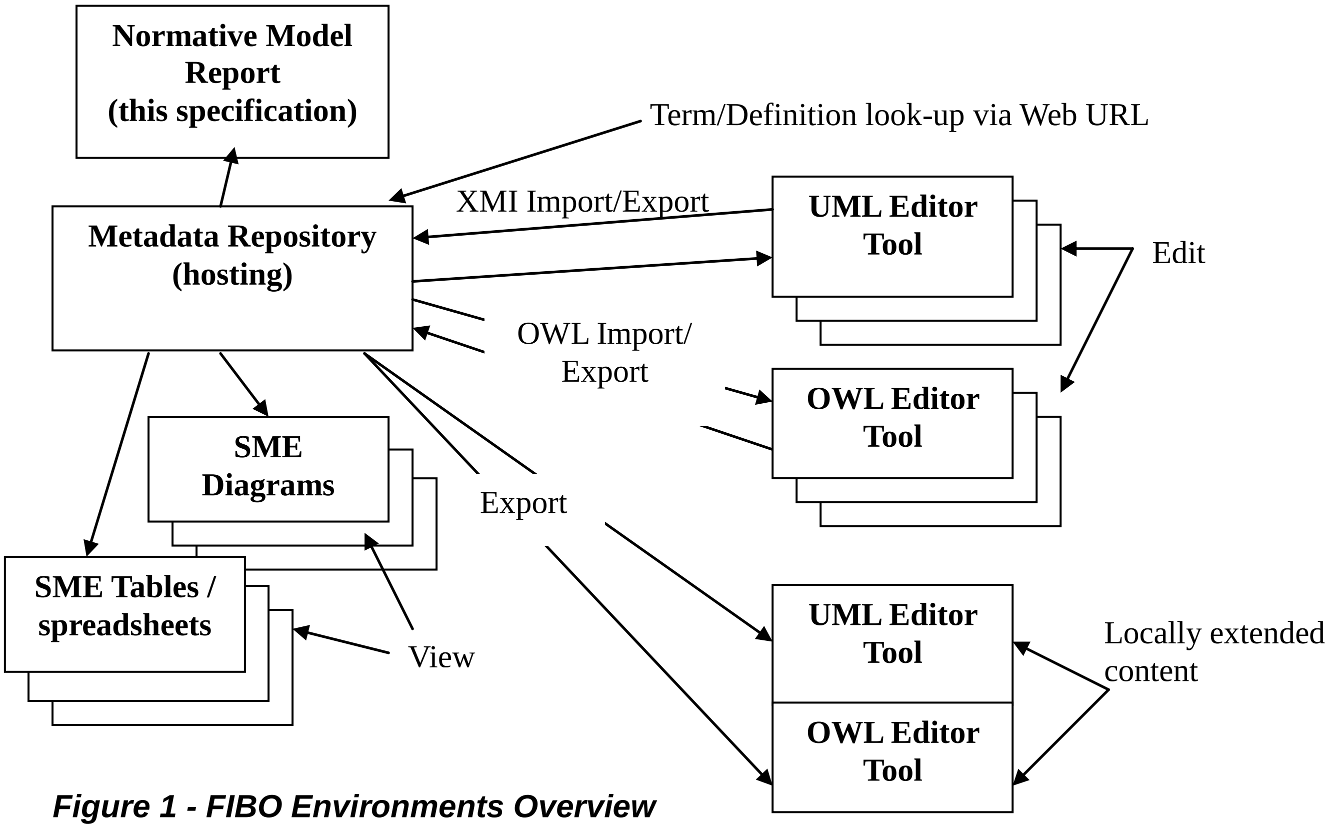


Figure 1 - FIBO Environments Overview

8.3 Ontology Definition Metamodel Usage and Adaptations

8.3.1 Introduction

The Ontology Definition Metamodel standard provides a means to represent OWL constructs within UML tools. This is achieved using a UML extension construct called a 'profile' for OWL and for RDF Schema within UML. The profile defines a number of UML base classes which may be used to represent OWL constructs in a consistent and meaningful way. The result of using the ODM specification is that one may render OWL models in a UML editor tool.

This specification takes ODM and explicitly defines a sub-set of it, which is to be used in the production of the diagrams described herein. Definition of that sub-set is a specific aspect of this specification, and is done in order to render diagrams which are suitable for business domain consumers or reviewers of the content of this specification.

In addition, this specification enhances these constructs with visual appearances (coloring of nodes and edges) so as to provide a visually richer appearance to the diagrams which are produced as described in this specification. The visual appearances themselves may not necessarily be represented in all renditions of the model content (for example in OWL or in different UML tools), and so do not form a normative element of this specification, however these are replicated here alongside the defined sub-set of ODM base classes, for completeness. In addition, most of the model content has appearances which are determined by the 'Archetypes' construct which is described in a separate section, and so only a limited number of these appearances (for example for OWL union classes) are seen in the final model content.

8.3.2 ODM Constructs Usage

Table 1 shows the RDF, RDF Schema and OWL model constructs, their corresponding UML base classes as used in this specification, the names of the stereotypes for the constructs and their appearances.

Table 1. ODM Constructs Usage

Construct Requirement	Stereotype	UML Base Class	Appearance
RDF Constructs			
Sub property	subPropertyOf	Generalization	Green vertical arrow
Sub-class	subClassOf	Generalization	Black vertical arrow or tree
Datatype	rdfsDatatype	Class	Green box
Instance type relationship	rdfType	Dependency	Brown dashed arrow
Cross reference	seeAlso	Dependency	Green dashed arrow
Comment	comment	Dependency	Green dashed arrow
Label	label	Dependency	Green dashed arrow
Is Defined By	isDefinedBy	Dependency	Green dashed arrow
Literal Data	rdfsLiteral	InstanceSpecification	Gray box
Typed Literal	typedLiteral	InstanceSpecification	Gray box
Plain Literal	plainLiteral	InstanceSpecification	Gray box
Instance of Annotation	fact	Dependency	Green dashed arrow
OWL constructs			
Class	owlClass	Class	Gold class box
Object Property	objectProperty	AssociationClass	Blue arrowed line with class box
Object Property	objectProperty	Attribute	Black text entry in class box*
Datatype Property	datatypeProperty	Attribute	Black text entry in class box
Union relation	unionOf	GeneralizationSet, defined as covering	Purple vertical arrow tree
Disjoint union relation	disjointUnionOf	GeneralizationSet, isCovering=True, isDisjoint=True	Purple vertical arrow tree
Intersection relation	intersectionOf	Generalization	Purple vertical arrow tree
Union Class	UnionClass	Class	Gold class box
Intersection Class	IntersectionClass	Class	Gold class box
Disjoint relation	disjointWith	Dependency	Red dashed arrow
Inverse relationship	inverseOf	Dependency	Red dashed arrow
Individual	owlIndividual	InstanceSpecification	Default
Named Individual	NamedIndividual	InstanceSpecification	Default
Anonymous Individual	AnonymousIndividual	InstanceSpecification	Default
OWL Annotation Property	annotationProperty	AssociationClass	Green arrowed line with class box
OWL Ontology	owlOntology	Package	Yellow package
Equivalent Class	equivalentClass	Dependency	Green dashed arrow
Same As	sameAs	Dependency	Green dashed arrow
Different From	differentFrom	Dependency	Green dashed arrow
Selection list	dataRange	Enumeration	Green enumeration class
Enumerated set	EnumeratedClass	Class	Gold class box
OWL Import	owlImports	Dependency	Light blue dashed arrow
Annotation instance	annotationFact	Dependency	Green dashed arrow

* The additional base class given for Object Property as a UML Attribute is provided for convenience in some models but is intended only to be used under certain defined conditions where the range of the object property is a basic, widely referenced class such as monetary amount.

8.3.3 Packaging

Model content is packaged for convenience into separate ontologies, rendered with the UML base class of 'Package' as shown above.

Disposition of the packaging within a given UML editor tool is not rendered in OWL representations of the FIBO model content. The thematic divisions represented by these "module" level UML packages are reflected in the namespace of the ontology as an additional namespace component. UML tools may or may not render this information as separate packages, and may or may not manage these as separate operating system files. Some conformance points may be stated with reference to these intermediate, "module" packages.

For the avoidance of doubt, the nesting of any package within any other package does not represent an implicit ontology relationship and no packages which are ontologies are or should be nested within any other package which is an ontology.

Relationships between ontologies (i.e. the OWL Import relationship) are rendered explicitly as OWL Import constructs in the content of the model described in this specification. No implicit imports are to be assumed which are not included in the FIBO model content.

8.4 The Global Terms Models

8.4.1 Rationale

As a consequence of the modeling principles, the model requires ontologies of things which are not specific to financial services or business entities. These include legal concepts like contracts, business concepts such as service provision, as well as an extensive set of concepts for times, dates, mathematical constructs, events and activities, and so on. It is for this reason that this ontology for business entities has been created to support financial industry business ontologies.

There are two important features to this part of the model:

1. These sections define the simplest or most generic kind of thing that something is (these are referred to as 'Archetypes');
2. In the future, these terms are to be defined with reference to known, proven standards in the industries for which they are defined or, in the case of non-industry specific concepts, some suitably well-referenced and adopted standard.

These terms are presented in a number of model sections, each containing a number of discrete ontologies. The content of these sections is further cross referenced to copies of such external ontologies as have been used as points of reference.

8.4.2 Archetypes

As defined in this specification, an archetype is simply the 'simplest kind of thing' for a particular kind of concept. For example 'Contract' represents the most basic form of contract, having the necessary facts which must be true of all things which are a contract. The term 'Contract' and the facts about it such as 'has principal' are all defined as archetypal classes of 'Thing' and archetypal relationship facts (OWL Object properties).

The description of an archetypal kind of thing and the set of necessary facts about that thing are referred to in diagram names as a 'Grammar'. The concept of 'Archetype' is in many ways similar to that of a stereotype in UML, with the important distinction that the archetype is also the highest level super-type of the things which share that archetype. In the example of Contract, all classes which refer to what are in actuality contracts, have the OWL class of 'Contract' as an ancestor. They therefore inherit all the facts which necessarily apply to contracts, except when these facts are defined by restriction of those archetypal facts, for example the fact that a security has an issuer is a restriction of the fact that a contract has a principal.

Archetypes are identified by some unique appearance in the form of a color or a graphic. The precise appearances of each archetype are not normatively defined in this specification but it is a requirement that all classes in the model (with the exception of OWL Union Classes and if used, OWL Intersection Classes) shall have an archetype and be represented in business diagrams in some unique way. An exception to this is the OWL classes used to represent the partitions described in the next section. This requirement does not extend to third party models derived by extension of this model, but it is strongly encouraged that people creating such models do retain the archetype distinctions if practicable.

8.5 Model Content Reporting

8.5.1 Model Visual Reporting

The model content may be presented to business domain experts in a number of formats, showing different levels of detail and different parts of the model content. The individual diagrams are not normatively defined in this specification. The basic requirements which must be met by such diagrams is normatively defined in this specification, as follows:

At least one type of diagram shall be produced, which is optimized for review by business domain experts. These diagrams shall require no knowledge on the part of those viewing them, of any formal modeling language or design techniques, and no knowledge of the Web Ontology Language or the names of the constructs thereof.

All visual elements of these diagrams shall be explainable with reference to established, non technical concepts. Such concepts may include set theory, basic Aristotelian logic and the like.

Figure 2 shows an example of one such diagram. Note that this is of a format which shows relationships between relationships. A version of each diagram in this format may also be created without the class icons for each relationship fact, for easier consumption by the business domain.

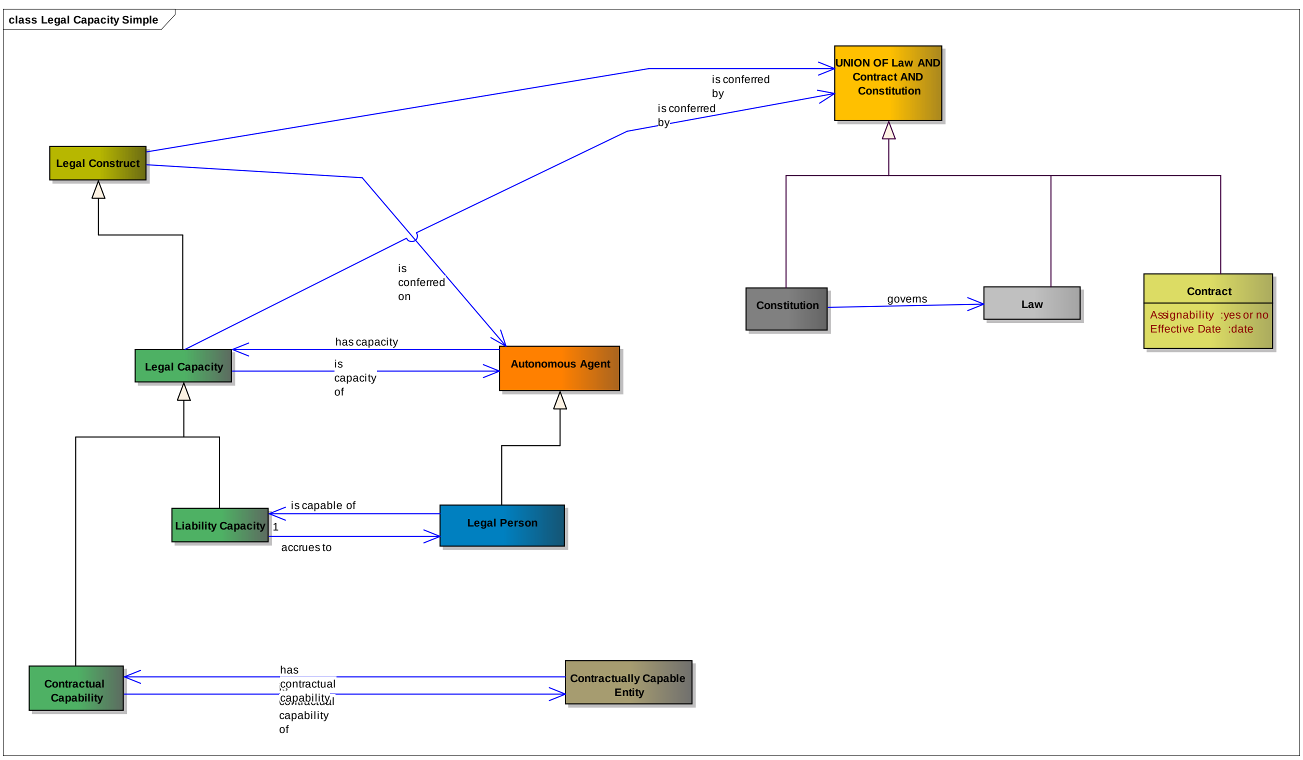


Figure2 : Example Business Diagram

8.5.2 Model Textual Reporting and Construct Naming

As with the visual display of model content by diagrams, there shall also be a set of tables provided, in tabular or spreadsheet format in the form of two-dimensional tables with column headings and with each row representing one meaningful concept.

There are two levels of detail which shall be made available in reports. These are the 'Basic' view of Term, Definition and Synonym, and an extended view giving most or all of the same information that is seen in the diagrams. This shall include line entries for each thing and each fact (relationship fact and simple fact) as well as the set theory constructs and relationships modeled (unions, parent terms etc.). It is not necessary to show relationships between relationships in these tables, such as sub property hierarchies or property inverses.

Each construct from which the model has been built shall be represented with an English language name as described in Table 2. These names are in US English and may be replaced in reports with definitionally equivalent labels in other human languages.

Table 2. ODM Constructs Appearances

Construct Description	Construct	English Name	Displayed when it appears in
RDF Constructs			
Sub property	subPropertyOf	Sub Property	Detail tables, detail diagrams
Sub-class	subClassOf	Is A	All tables, diagrams
Datatype	rdfsDatatype	Type	No diagrams, no tables
Type instance relationship	rdfType	type of	No diagrams, no tables
Cross reference	seeAlso	See also	Annotation reports, annotation diagrams only
Comment	comment	Comment	Annotation reports, annotation diagrams only
Label	label	Lexical Label	Annotation reports, annotation diagrams only
Is Defined By	isDefinedBy	Defined by	Annotation reports, annotation diagrams only
Literal data	rdfsLiteral	Annotation content	Annotation reports, annotation diagrams only
Typed literal	typedLiteral	Typed Literal	Annotation reports, annotation diagrams only
Plain Literal	plainLiteral	Plain Literal	Annotation reports, annotation diagrams only
Instance of annotation	fact	Fact	Annotation reports, annotation diagrams only
Subject of instance	subject	subject	Annotation reports, annotation diagrams only
Predicate of instance	predicate	predicate	Annotation reports, annotation diagrams only
Object of instance	object	object	Annotation reports, annotation diagrams only
OWL constructs			
Class	owlClass	Thing	All tables, diagrams
Object property	objectProperty	Relationship Fact	all tables, diagrams
Datatype Property	datatypeProperty	Simple Fact	All tables, diagrams except block
Union relation	unionOf	union of	All tables, diagrams
Disjoint union relation	disjointUnionOf	mutually exclusive union of	All tables, diagrams
Intersection relation	intersectionOf	intersection of	All tables, diagrams
Union Class	UnionClass	Union	All tables, diagrams
Intersection Class	IntersectionClass	Intersection	All tables, diagrams

Construct Description	Construct	English Name	Displayed when it appears in
Disjoint relation	disjointWith	mutually exclusive	Detail tables, all diagrams
Inverse relationship	inverseOf	inverse	Detail diagrams only
Individual	owlIndividual	Individual	All tables, diagrams
Named Individual	NamedIndividual	Named Individual	All tables, diagrams
Anonymous Individual	AnonymousIndividual	Anonymous Individual	All tables, diagrams
OWL Annotation Property	annotationProperty	Annotation Type	Annotation Reports, Annotation diagrams only
OWL Ontology	owlOntology	Ontology	Ontology relations diagrams, no tables
Equivalent Class	equivalentClass	Equivalent Thing	Ontology relations, provenance diagrams, no tables
Same As	sameAs	Same Thing	Ontology relations, provenance diagrams, no tables
Different From	differentFrom	Different Thing	Ontology relations, provenance diagrams, no tables
Selection of values	dataRange	Selection	All diagrams; separate tables
Selection of Classes	EnumeratedClass	Selection of Things	All tables, diagrams
OWL Import	owlImport	Ontology Import	Ontology relations diagrams, no tables
Annotation instance	annotationFact	Label according to the type of annotation this is	Annotation Reports, Annotation diagrams only

9 Additional Metadata

9.1 Introduction

The model is supported by additional metadata. These cover features which are not part of the OWL language (and therefore not in ODM) but which are necessary additional annotations to the constructs in the model. This section describes what metadata is provided for in the model and how it is rendered.

9.2 Metadata Types

Metadata is provided for the following separate reasons, and is described in separate headings according to those reasons:

- Basic Annotation
- Provenance and cross reference annotation
- Definition and additional notes annotation
- Contextual annotation
- Change management annotation

9.2.1 Basic Annotation

This covers aspects of model elements (classes and relationships) which are not provided for in the OWL language. These are:

- Synonyms
- Archetypes

Synonym

Synonyms are fundamental to the reporting required for business domain view and review of the model content, which requires term, definition and synonym, and in many cases nothing more.

A fundamental principal of this model is that it is an ontology and not a vocabulary or terminology. For this reason, the model contains, and models derived from it should contain only one class per single concept. The use of separate classes with the same meaning, and the use of the OWL construct for class equivalence (equivalentClass) shall not be used except when stating equivalences between classes in different ontologies, different named graphs or any other context in which the same concepts may exist in different namespaces. Instead, for each concept, any additional names by which that concept may be referred shall be represented as synonyms.

Archetype

The concept of archetypes is not part of the OWL language, and is a unique and novel aspect of the model described in this specification. Each class and object property is identified with an archetype. In UML representations these are mechanized as UML stereotypes. In order to preserve the archetype information in OWL models, these are rendered as OWL Annotation Properties.

9.2.2 Provenance and Cross-reference Annotation

Information is maintained in the model for the origin of each term and definition, including definitions which are adapted from a given source rather than being a direct rendition of that definition.

Similar terms are used for cross reference to terms and definitions in other standards or sources. These are similar to the provenance terms but they do not represent the origin of the term or definition.

Term Provenance meta-terms are all derived from the Dublin core construct called 'source'.

There are two types of meta-term for the origins of terms and definitions:

- Term Origin
- Definition Origin

These are further refined as follows:

Term Origin:

- The source of the term
    - In a standard or draft standard;
    - In some other document;
    - Provided by some organization;
- The name of the term in the source (typically the name of a UML data element or an XML schema construct)

Definition Origin:

The definitions are either replicated directly from the originating source (if intellectual property considerations permit this), or are adapted from these. Adapted definitions are typically created because the definition in the originating model or source is a definition of a data element or an XML Schema construct and not a definition of the real world entity to which that construct relates.

This leads to two separate definition origin related meta-terms:

- Definition Origin - used where the text in the skos:definition (the main definition field in this model) is a direct copy of the definition of the term defined in the Term Origin meta-terms

- Definition Adapted From - used where the text in the skos:definition is a modified rendition of the text of the term defined in the Term Origin meta-terms.

Table 3 shows the metadata used.

Table 3. Provenance and Cross Reference Metadata

Base Term	Annotation (meta-term)	Target term (range)	Notes on Usage
Dublin Core			
dct:source	TermOriginDocument	Document	The document (potentially including standard document) from which the term was sourced
dct:source	TermOriginStandard	Standard	The standard from which the term was sourced
dct:source	TermOriginalName	Text literal	The name of the term in the original source
dct:source	DefinitionOrigin	Document	The work from which the definition was sourced
dct:source	DefinitionAdaptedFrom	Document	The work from which the definition was modified.

Note that DefinitionOrigin and DefinitionAdaptedFrom are mutually exclusive. There is no logic to enforce this.

9.2.3 Definition and Additional Notes Annotations

Annotations for the formal definition of each term, and for additional notes are derived from the Simple Knowledge Organization System (SKOS) standard. During creation of the original model these elements of text were retained in the UML 'Notes' field.

Notes Annotations

The following terms exist in SKOS as specializations of the SKOS element skos:note:

- skos:definition
- skos:editorialNote
- skos:scopeNote
- skos:historyNote
- skos:example
- skos:changeNote

The terms previously maintained as part of the definition and notes text in the UML models are split into one or more of the above SKOS annotations. Of these, skos:definition must always be present, while the remaining terms may or may not be populated. Note that the earlier development stages of the model described in this specification, which were done in a UML modeling tool, had the definition along with a set of 'Further Notes' in the UML 'Notes' model element.

In addition, the following terms are defined in SKOS as specializations of the RDF element Label:

- altLabel
- prefLabel
- hiddenLabel

These may be used as they stand. In addition, two extension terms are defined for skos:altLabel:

- abbreviation
- operationalLabel

Cross Reference Annotations

Standards in the "Global terms" section are formally cross referenced to ontologies or standards which have the same meaning and which have been selected as being the place of record for the meaning of a given term. Usually these are ontologies, and are referenced using OWL annotations for class equivalence. In some cases the resource to which we want to cite the meaning of a term is in some other format such as UML, and in this instance an additional annotation element is used, which is "citation". The citation metadata construct is defined as a sub-type of the built in vocabulary element "isDefinedBy" which is a sub-type of the RDF element "seeAlso".

Table 4 shows the SKOS-derived annotations plus the citation annotation construct.

Table 4. Labeling, Notes and Cross Reference Metadata

Term Requirement	Term Type	Annotation (meta-term)	Notes on Usage
SKOS Notes		skos:note	
Definition	Definition	skos:definition	Main formal definition of term
General notes	Notes	skos:editorialNote	The bulk of the 'Further Notes' narrative
Scope Note	Notes	skos:scopeNote	Additional formal information about the term or concept
Historical Note	Notes	skos:historyNote	Notes from historical review sessions
Example	Notes	skos:example	Previously in UML Notes
Usage Note	Notes	skos:note	Previously in UML Notes
SKOS Labels			
Preferred Label	Labels	skos:prefLabel	Main label in US English
Alternate Label	Labels	skos:altLabel	Synonym
Change History	Notes	skos:changeNote	Part of change control terms
SKOS Extensions			
Abbreviation	Labels	abbreviation	Alternative abbreviation for term
Operational labels	Labels	operationalLabel	Use for operational ontologies
RDF Built-In Terms			
Semantics Cross ref.	Sub-type of RDF isDefinedBy	citation	Citation where source is not OWL

9.2.4 Contextual Annotation

The model includes metadata for deriving extracts from the model content for specific applications, both conventional and semantic web.

Context is defined by the use of OWL Object Properties (relationships facts) with a range that is some term derived from the 'Mediating Thing' class. These terms are the business contexts which have been modeled in this model. These contexts, and the relationships which refer to them, are not shown on most diagrams but are to be included on diagrams which show the origins and cross references of terms.

One additional metadata requirement for context is the ability to identify, for a given set of sub-classes of a given class, what was the property or properties of the parent class which is restricted or specialized to derive that set of sub-classes. This has important applications in the extraction of model content both for model driven development and for semantic technology applications.

The 'Classification Facet' metadata formally identifies a set of terms which are mutually exclusive to one another and which share a single parent. Optionally, the Classification Facet further relates the set of terms to the property by which they are specialized, to a 'Context' class of thing ('third order thing'). Therefore the metadata has a range which is either an object property or an OWL class.

This metadata is rendered in much the same way as the other annotation metadata: it is rendered in OWL as an OWL Annotation Property (stereotype annotationProperty), and rendered in UML as an Association Class with a green relationship line (edge). Instances of the type of annotation which is a Classification facet are shown as a green dependency edge, and in the UML rendition these may have a range either of the class element of the Association Class for an Object Property, or of a UML class which represents some OWL Class.

9.2.5 Unique Metadata and Annotations

The following meta-terms are introduced as part of this specification and are not derived from other terms or standards:

- Archetype terms
    - Term identifying something as an archetype
    - Annotation indicating what archetype a given class is of.
- Classification Facet

These are given in Table 5.

Table 5. Unique FIBO Metadata

Term Requirement	Annotation (meta-term)	Rendition	Notes on Usage
Archetype			
Class or Object Property is an Archetype	archetypal	Instance points to RDFS type literal of type "boolean" and set to 'yes'	Annotation of Class (boolean)
Class is of archetype	ofArchetype	ofArchetype	Relates class or object property to the class or object property which is its archetype
Classification Facet			
Type of annotation is Classification Facet	none	UML AssClass / OWL Annotation Property	Defined once. Range is union of Class and Object Property
The concept of a Classification Facet	isClassificationFacet	unionClass	The class represents the classification facet itself, and can be further related to things in the model (e.g. context)
Instance of Classification Facet	inClassificationFacet	UML Dependency / OWL annotationFact	Range is the Classification Facet class.

9.2.6 Change Management Annotation

Annotation for change management is derived directly from the OMG AB Recommendation for ontologies metadata and it not re-specified here.

The formal version information for each element is given using the OWL construct owlVersionInfo

Notes made as part of the change management process (change notes etc.) are rendered using the SKOS element skos:changeNote as listed in the preceding section.

9.3 Metadata Rendition

The additional metadata described in the preceding section is rendered as OWL Annotation Properties.

Note that in ODM both RDF and OWL have a construct with the stereotype of 'annotationProperty'. The one used for metadata here is the OWL Annotation Property construct.

The metadata terms are defined wherever possible as extensions of RDF and OWL terms or of Dublin Core (DC) and (Simple Knowledge Organization System (SKOS) terms. These are replicated in the model repository and from these terms is created a set of sub-terms which define the OWL Annotation Properties that will be taken to represent those terms.

The OWL Annotation Properties for, for example, Definition Origin, represent a type of annotation (in this example, the origin of a definition), and these model at a class level what sort of things may be the domain and range of the annotation property. Individual instances of those annotation properties are annotation facts, and these are accordingly modeled as the owl term annotationFact. This is not to be confused with the RDF term also known as annotationFact in the ODM standard.

An annotation fact (stereotype annotationFact) is rendered as a UML Dependency. Annotation facts are instances of annotation properties. For each type of metadata term which is defined here as an OWL annotation property, there is a corresponding annotation fact which is defined as being an instance of that type of property (for example, an instance of the type of property which is a definition origin annotation).

All semantic provenance and cross reference metadata is rendered visually as green relationships. These are intended to be displayed on diagrams drafted explicitly to show this metadata and are not intended to be visible in business-facing diagrams which show only the things and facts. Tabular reports may include or not include this information.

10. Model Content Reports

10.1 Overview

Please note that the content of the Global Terms ontologies is not presented as financial industry content and exists only to support the content in other FIBO specifications. Terms in those specifications shall make normative reference to the relevant terms in this specification.

This section shows all content in the model. Note that the annotations to the model (definitions, editorial notes and the like) are maintained in the model as described in Section 9 but for convenience these are reported here as textual annotations. The name of the annotation is given in bold, and the literal text content of that annotation is shown as the text which follows the annotation.

10.1.1 Interpreting This Section

This section shows each of the components of the model with their OWL construct names where applicable. These are:

Construct Name	Description
Model Section:	A grouping of ontologies with some common theme. These also share a namespace fragment in the corresponding OWL files.
owlOntology	A single OWL ontology.
owlClass	An OWL Class, that is a set theoretic construct representing a common set of properties, possession of which would make any individual a member of this set.
owlObjectProperty	The Class named as “Range” for the relationship represents something in terms of which the meaning of the relationship is framed. Known as “Relationship fact” in business spreadsheets.
rdfsSubClassOf	**“is a”** relationships - these have no definition. This relationship indicates that the Class is a sub-class of the Class named as the “Range” in the relationship. Known as “Parent” in business spreadsheets.
owlDatatypeProperty	Some property framed in terms of some simple type of information such as text or a “yes or no” value. Known as “Simple Fact” in business spreadsheets.
owlDatatypeProperty Range	The type of information in which the OWL Datatype Property is framed Known as “Simple Type” in business spreadsheets. NOTE: for some datatype properties, the range is a DataEnumeration (see below). NOTE: For some datatype properties, the fact type is given as a Class e.g. Monetary Amount. In such cases, this is intended to be an OWL Object Property. The use of this style of object property is a convenience for diagrams production. This will be corrected in future versions of this specification.
DataEnumeration	These item represent a selection of possible values, which are intended to be taken as literal (e.g. textual) values. A “Simple Fact” (OWL Datatype Property) may identify one of these as the Simple Fact Type; this means that any one of the values in the list may be a possible value for this property.

Construct Name	Description
UnionClass	This corresponds to a logical union of Classes. The membership of the union is not shown in this report.
disjointWith	Identifies two sets of which no one individual may be a member of both. Known as "mutually exclusive" in business spreadsheets.
Definition	The SKOS Definition annotation, giving the formal definition of the item
Editorial Note	The SKOS Editorial Note annotation, giving additional narrative about the term and definition. Includes line breaks and additional narrative headings within this annotation, i.e. everything up to the next annotation or construct entry is part of this annotation.
Scope Note	The SKOS Editorial Note annotation, giving notes about the scope and application of the term.
Term Origin	A temporary annotation, to be replaced by a range of FIBO-specific annotations derived from the Dublin Core "source" property. These will include: TermOriginDocument TermOriginStandard TermOriginalTerm
Definition Origin	A temporary annotation, to be replaced by a range of FIBO-specific annotations derived from the Dublin Core "source" property. These will include: DefinitionOrigin DefinitionAdaptedFrom
Consensus	An annotation from the EDM Council working sessions, this will not be included in the formal submission of this specification and these will be removed.

10.2 Model: Vocabulary, Annotation and Other Supporting Terms

10.2.1 Model Section: Ontology Vocabulary

10.2.1.1 Ontology: BusinessTypes

URI

http://www.omg.org/spec/FIBO/Foundation/Types/BusinessTypes/

10.2.2 Model Section: Annotation Ontologies

10.2.2.1 Ontology: ProvenanceAnnotation

URI

http://www.omg.org/spec/FIBO/Foundation/Annotation/ProvenanceAnnotation/

Classes

Class: Draft ISO Standard

Parents

- Draft Standard

Class: Draft Standard

Parents

- Technical Standard

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Document Identifier		text		No	
Draft Version		text		No	

Class: Published ISO Standard

Parents

- Published Standard

Class: Published Standard

Parents

- Technical Standard

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Publication Date		date		No	
Version		text		No	

10.2.2.2 Ontology: FIBOArchetype

URI

http://www.omg.org/spec/FIBO/Foundation/Annotation/FIBOArchetype/

10.2.3 Model Section: External Utility Ontologies

10.2.3.1 Ontology: W3CTime

URI

http://www.omg.org/spec/FIBO/Foundation/ext/snap/partial/W3CTime/

Classes

Class: Instant

Parents

- Temporal Entity

Mutually Exclusive Classes

Interval

Class: Interval

definition: A Proper Interval, i.e. an Interval whose beginnign and end are different Note: we have ignored the distinction in the W3C suggested OWL ontology for Time, which has Interval and Proper Interval as separate types of Thing.

Parents

- Temporal Entity

Class: Temporal Entity

10.2.4 Model Section: Global Standards Semantics

10.2.4.1 Model Sub-section: Snapshots

10.2.4.1.1 Ontology: UN-FAOCountry

URI

http://www.omg.org/spec/FIBO/Foundation/ext/snap/partial/UN-FAOCountry/

Classes

Class: DisputedTerritory

Parents

- GeopoliticalTerritory

Class: EconomicRegion

Parents

- GeopoliticalGroup

Class: GeographicalRegion

Parents

- GeopoliticalGroup

Class: GeopoliticalEntity

Class: GeopoliticalGroup

Parents

- GeopoliticalEntity

Class: GeopoliticalOrganization

Parents

- GeopoliticalGroup

Class: GeopoliticalSpecialGroup

Parents

- GeopoliticalGroup

Class: GeopoliticalTerritory

Parents

- GeopoliticalEntity

Class: NonSelfGoverningTerritory

Parents

- GeopoliticalTerritory

Class: OtherTerritory

Parents

- GeopoliticalTerritory

Class: SelfGoverningTerritory

Parents

- GeopoliticalTerritory

10.3 Model: FIBO-Foundation

10.3.1 Model Section: GoalsAndObjectives

10.3.1.1 Ontology: Goal

URI

http://www.omg.org/spec/FIBO/Foundation/GoalsAndObjectives/Goal/

Classes

Class: Desired Result

Class: Goal

isArchetype: true

Parents

- Desired Result

10.3.2 Model Section: AgentsPeople

10.3.2.1 Ontology: Agent

URI

http://www.omg.org/spec/FIBO/Foundation/AgentsPeople/Agents/

Classes

Class: Autonomous Agent

isArchetype: true

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
has capacity (in ontology LegalCapacity)		Legal Capacity		Yes	is capacity of
formerly known as (in ontology EntityName)	**definition:** A name by which the entity has been known in the past but is not known by at the present time.	Name	called	Yes	
called (in ontology EntityName)	**definition:** The name by which the autonomous thing is known.	Name	details	Yes	

Class: Enabling Agent

definition: Something which is capable of enabling some Event or providing some service.
editorial note: This is defined as a Relative Thing, that is something is only regarded as being a kind of agent in the context in which it is the agent of something - nothing is simply an "Agent" in vacuo. Editorial Note: Since the term "Agent" is frequently used in technical applications to mean some independent thing which may be an agent in some context (usually the context implicit in the application), we have renamed this term to Enabling Agent. This allows us to retain some fidelity to the English language while avoiding confusion caused by the labels typically chosen by data modelers. Scope Note: This term is the abstraction from which are drawn various kinds of service providers i.e. business acting or capable of acting as agents in the provision of some service.

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
has identity	**definition:** The thing which is able to perform the role of Agent.	Thing		Yes	

10.3.2.2 Ontology: Person

URI

http://www.omg.org/spec/FIBO/Foundation/AgentsPeople/Person/

Classes

Class: Adult

definition: Definition needed

Parents

- Person

Class: Child

definition: Definition needed

Parents

- Person

Class: Emancipated Minor

definition: Definition needed

Parents

- Minor

Class: Incapacitated Adult

definition: Definition needed

Parents

- Adult

Class: Minor

Parents

- Person

Mutually Exclusive Classes

Natural Person

Class: Person

definition: A person; any member of the species homo sapiens.

Parents

- Autonomous Agent

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Human Being has legal name (in ontology EntityName)	**definition:** The legal name by which the person is known.	Personal Name	called	Yes	
Date Of Birth		date		No	
Gender		Gender		No	

Enumerations

Enumeration: Gender

Annotation

Allowed Values

- Male
- Female
- Unspecified

10.3.2.3 Ontology: Organization

URI

http://www.omg.org/spec/FIBO/Foundation/AgentsPeople/Organization/

Classes

Class: Organization

isArchetype: true

Parents

- Autonomous Agent

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
has address (in ontology OrganizationAddress)	**definition:** The address at which the entity is known and may receive correspondence.	Postal Address	details	Yes	
has goal (in ontology Goal)	**definition:** The goal or objective for which the Organization was set up and which is the reason for its existence. This may be to make a profit for shareholders, to discharge the responsibilities of a Government, to propagate a faith or political purpose etc.	Goal		Yes	
has organization member (in ontology OrganizationMember)	**definition:** An entity which is a member of the organization.	Organization Member		Yes	

Name	Annotation	Type	Parent	Multiples	Inverse
has member (in ontology OrganizationPart)	**definition:** The individuals that make up the organization.	Autonomous Agent		Yes	
has organization part (in ontology OrganizationPart)	**definition:** Some part of the organization, which is itself an organization.	Organization		Yes	

10.3.3 Model Section: SocialConstruct

10.3.3.1 Ontology: SocialConstructsCore

URI

http://www.omg.org/spec/FIBO/Foundation/Social/SocialConstructsCore/

Classes

Class: Social Construct

isArchetype: true

10.3.3.2 Ontology: Control

URI

http://www.omg.org/spec/FIBO/Foundation/Social/Control/

Imports

- SocialConstructsCore

Classes

Class: Control

definition: The term “control” (including the terms “controlling”, “controlled by” and “under common control with”) means the possession, direct or indirect, of the power to direct or cause the direction of the management and policies of a person, whether through the ownership of voting shares, by contract, or otherwise.

Parents

- Social Construct

Class: De Facto Control

Parents

- Control

Mutually Exclusive Classes

De Jure Controlling Interest

Class: De Jure Controlling Interest

definition: Control which is mechanized by some formal legal construct.

Parents

- Control
- Legal Construct

10.3.3.3 Ontology: Agreements

URI

http://www.omg.org/spec/FIBO/Foundation/Social/Agreements/

Classes

Class: Agreement

editorial note: An Agreement may be formalized in the form of a Contract or other formal instrument, or it may not. In either case, the agreement is that which may be referred to as the agreement between or among the parties, and the contract is framed as defining (and usually as exclusively defining) the agreement between two parties. This is framed as an Independent Thing, that is the agreement itself. A relative thing would be "the agreement between...".
definition: Some mutual undertaking or set of undertakings between two or among several parties.

Class: Bilateral Agreement

definition: Some Agreement which is between two parties.

Parents

- Agreement

10.3.3.4 Ontology: Ownership

URI

http://www.omg.org/spec/FIBO/Foundation/Social/Ownership/

Classes

Class: Owner

definition: A party which owns something. The thing owned is an Asset to that Party.

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
owner has identity	**editorial note:** In general this may be any entity which is capable of holding and owning any type of ownership instrument. At this most general level, it is considered as being anything at all. This should be restricted or constrained in some way for specific contexts of ownership such as share ownership, since at this level there is no such constraint (it may be a gorilla owning a toy). **definition:** That which may perform the role of Owner.	Autonomous Agent		Yes	
owns	**definition:** A thing owned by the party.	Thing		Yes	

Class: Ownership

definition: Ownership is the context in which some Party is said to own some Independent Thing. The Party is defined as such due to its being the owning party to that Thing.

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
involves	**definition:** The context of Ownership involves some party which is defined as an Owner.	Owner		Yes	

Class: Physical Asset

definition: A Physical Thing held by some party and having some value.

Parents

- Asset

Class: Asset

definition: A Thing held by some party and having some value.

Properties

Name	Annotation	Type	Parent	Multiples	Inverse

Name	Annotation	Type	Parent	Multiples	Inverse
Asset owned by	**definition:** The party which is the Owner of the Asset. **editorial note:** It is part of the definition of an Asset that it is owned by some Owner.	Owner		Yes	
Asset takes form of	**definition:** The form of the thing which is held as an asset.	Thing		Yes	
measured in		Currency		No	

Class: Property Asset

definition: A physical property (building) regarded as an asset.

Parents

- Physical Asset

10.3.4 Model Section: Legal

10.3.4.1 Ontology: LegalCore

URI

http://www.omg.org/spec/FIBO/Foundation/Legal/LegalCore/

Classes

Class: Constitution

isArchetype: true

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
governs	**definition:** The constitution governs and constrains the application of the or each body of law in the jurisdiction other framework to which it applies. **editorial note:** These relations are also general to non national equivalents where for example the constitution is the company articles and the law is the bylaws of the company.	Law		Yes	

Class: Court Of Law

definition: Some formal body with the mandate to hear disputes or cases.
editorial note: temporary definition.

Parents

- Formal Organization

Class: Law

isArchetype: true

Class: National Constitution

editorial note: This defines the framework in which are made and in which they have force.
definition: A body of rules, commitments and statements that set out how a country, state or territory is to be run.

Parents

- Constitution

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
constrains (in ontology Jurisdiction)	**definition:** The Constitution constrains what can or cannot be legislated as Law for the Jurisdiction in which that Constitution applies.	Statute Law	governs	Yes	
governs Legal System (in ontology Jurisdiction)	**definition:** The legal system defined in and described by the National Constitution.	Legal System		Yes	

Class: Ordinance

definition: An authoritative rule or law; a decree or command; a public injunction or regulation, such as a city ordinance against excessive horn blowing. (Source: Dictionary.com)

Parents

- Law

10.3.4.2 Ontology: Jurisdiction

URI

http://www.omg.org/spec/FIBO/Foundation/Legal/Jurisdiction/

Imports

- Statute

- LegalCore

Classes

Class: Bijuridicial Jurisdiction

editorial note: Examples include South Africa, Quebec, Louisiana and a few others, where or example Civil Law and Common Law arrangements are in force.
definition: A Jurisdiction in which two or more systems of law are in force.

Parents

- Jurisdiction

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Bijuridicial Jurisdiction partially governed under Civil Law System	**definition:** One system of law under which a Bijuridicial Jurisdiction is governed is the system of Civil Law. **editorial note:** A Bijuridicial Jurisdiction is governed under two systems of law, of which this is one.	Civil Law System	Jurisdiction governed under	Yes	
Bijuridicial Jurisdiction partially governed under Common Law System	**editorial note:** A Bijuridicial Jurisdiction is governed under two systems of law, of which this is one. **definition:** One system of law under which a Bijuridicial Jurisdiction is governed is the system of Common Law.	Common Law System	Jurisdiction governed under	Yes	

Class: Body Of Law

definition: Some body of law, that is some set of laws, statutes, ordinances and the like.

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
incorporates some	**definition:** The laws included in this Body of Law.	Law		Yes	

Class: Canon Law System

definition: The body of laws & regulations made or adopted by ecclesiastical authority, for the government of the Christian organization and its members.
editorial note: In force in both the Western (Roman) and Eastern churches, and in some form in the Anglican Church. There appear to be no current instances national jurisdictions which take this as a legal framework, with the possible exception of the Vatican. In historical contexts there would be.

Parents

- Religious Law System

Class: Civil Law

definition: The branch of the law dealing with disputes between individuals or organizations, in which compensation may be awarded to the victim.

Parents

- Common law

Class: Civil Law Jurisdiction

definition: A civil law jurisdiction.

Parents

- Jurisdiction

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Civil Law Jurisdiction governed under	**definition:** The system of law under which a Civil Law Jurisdiction is governed. This is the system of Civil Law.	Civil Law System	Jurisdiction governed under	Yes	

Class: Civil Law System

definition: A system of law based on Roman Law.

Parents

- Legal System

Class: Common Law Jurisdiction

definition: A jurisdiction based on common law.

Parents

- Jurisdiction

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Common Law Jurisdiction governed under	**definition:** The system of law under which a Common Law Jurisdiction is governed. This is the system of Common Law.	Common Law System	Jurisdiction governed under	Yes	

Class: Common Law System

definition: A system of law whose sources are the decisions made by judges.
editorial note: A jurisdiction which is based in Common Law will also have alongside a legislature that passes statutes.

Parents

- Legal System

Class: Common law

definition: Law developed by judges through decisions of courts and similar tribunals rather than through legislative statutes or executive branch action.

Parents

- Body Of Law

Class: Constitutional Law

definition: The body of law which defines the relationship of different entities within a state, namely the executive, legislature and the judiciary. Further notes: (from Wikipedia): Constitutional laws may be considered second order rulemaking or rules about making rules to exercise power.

Parents

- Body Of Law

Class: Criminal Law

definition: The body of law that defines conduct that is not allowed because it is held to threaten, harm or endanger the safety and welfare of people, and that sets out the punishments to be imposed on people who do not obey these laws.

Parents

- Common law

Class: International Law

definition: The set of rules generally regarded and accepted as binding in relations between states and nations.
editorial note: This can refer to three distinct legal disciplines: 1. Public international law governing the relations between provinces and national entities 2. Private international lse, which addresses questions of which jurisdiction may hear a case and which jurisdiction applies to issues in a case 3. Supranational law which concerns regional agreements where the laws of nation states may be held inapplicable when conflicting with a supranational legal system.

Parents

- Body Of Law

Class: Jurisdiction

isArchetype: true

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Jurisdiction governed under		Legal System		Yes	Legal System applies in
has in force (in ontology Statute)		Statute Law		Yes	is in force in
Full Name		text		No	
Common Name		text		No	

Class: Legal System

editorial note: This is a Mediating Thing, that is some context in which things have their meaning and existence - in this case, laws and the interpretation thereof by courts.
definition: Some system of Law.

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Legal System applies in	**definition:** The jurisdiction which is governed under the system of law.	Jurisdiction		Yes	

Name	Annotation	Type	Parent	Multiples	Inverse
defines the application of	**definition:** The Body of Law, the application of which is defined in the Legal System.	Body Of Law		Yes	

Class: Religious Law

definition: Some set of laws derived from and practised in the name of some religion.

Parents

- Body Of Law

Class: Religious Law Jurisdiction

definition: A Jurisdiction in which some system of religious law is in force.
editorial note: This includes Sharia Law jurisdictions.

Parents

- Jurisdiction

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Religious Law Jurisdiction governed under	**editorial note:** There are several mutually incompatible systems of Religious Law - the term referred to here is ancestral to all of those and defines simply that the jurisdiction in question is some kind of religious law jurisdiction - specific types are to be defined per system of religious law (two are included in the model for reference and possible use; there may be others). **definition:** The system of law under which a Religious Law Jurisdiction is governed. This is some system of Religious Law.	Religious Law System	Jurisdiction governed under	Yes	

Class: Religious Law System

definition: Some system of law based on the ordering principle of reality as being knowledge as revealed by God and governing all human affairs.

Parents

- Legal System

Class: Sharia Law Jurisdiction

definition: A Jurisdiction in which a Shari'a system of law is in force.

Parents

- Religious Law Jurisdiction

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Sharia Law Jurisdiction governed under	**definition:** The system of law under which a Shari'a Law Jurisdiction is governed. This is the system of Shari'a Law.	Sharia Law System	Religious Law Jurisdiction governed under	Yes	

Class: Sharia Law System

definition: A system of Shari'a Law.
editorial note: There are several schools of law in Islamic jurisprudence; these are not articulated here at present. The bodies of law and other textual corpora referred to in the implementation of this system of law are the Qur'an and the Hadith, with variations in interpretation and precedence according to the system (school) in question. Also known as the Qanun 'Islami (Islamic Canon)

Parents

- Religious Law System

Class: Statutory Law

definition: Written law set down by a legislature or by a legislator.
editorial note: Statutory Law may originate with national, state legislatures or local municipalities. Statutes of lower jurisdictions are subordinate to the law of higher. Definition adapted from: Wikipedia

Parents

- Body Of Law

10.3.4.3 Ontology: LegalConstructs

URI

http://www.omg.org/spec/FIBO/Foundation/Legal/LegalConstructs/

Imports

- Contract
- LegalCore
- SocialConstructsCore

Classes

Class: Commitment

editorial note: The undertaking by some party to act or refrain from acting results in an obligation on the part of that party, and usually results in the existence of some corresponding right on the party of some other party, in the event that the commitment is to such party. Thus Obligations and Rights are considered as reciprocal aspects of this Commitment concept.
definition: A legal construct which represent the undertaking on the part of some party to act or refrain from acting in some manner.

Parents

- Legal Construct

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
mandated by	**definition:** That which mandates the commitment i.e. the instrument or means through which that commitment exists.	UNION OF Law AND Contract AND Constitution	is conferred by	Yes	

Class: Duty

editorial note: This can also be thought of as an obligation - not in the sense in which an obligation and a right are the converse aspects of one another, but in and of itself, independent of the perspective from which it is considered. Examples include statutory obligations, reporting obligations and so on.
definition: Some obligation which exists and is imposed on some individual.

Parents

- Legal Construct

Class: Economic Commitment

definition: Some Commitment which forms part of the subject of some Transaction, being an undertaking by one or other of the parties to the transaction, extended to the other party to that same transaction.

Parents

- Mutual Commitment

Class: Mutual Commitment

definition: A commitment undertaken as between two parties.

Parents

- Commitment

Class: Statutory Responsibility

definition: An obligation which is defined under some body of law (statute).

Parents

- Duty

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
mandated by Statute Law	**definition:** The obligation is mandated under some body of statute law. **editorial note:** Note that the relevant body of law will also have terms (not shown) for identifying the nature of the entity that comes under the purview or mandate of this set of laws.	Statute Law	is conferred by	Yes	

Union:

Annotation

Union Of

Class: Uniteral Commitment

definition: A commitment made by one party without reference to the party to which the commitment is made.

Parents

- Commitment

Class: Legal Construct

editorial note: Obligations are an aspect of this category of thing, as are rights.
definition: Something which is conferred by way of law or contract, such as a right.

Parents

- Social Construct

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
is conferred on	**definition:** That to upon which the conferred thing is conferred. **editorial note:** The Conferred thing may be a right, a duty, an obligation, some legal capacity and so on, and is typically constrained to some sub-set of the whole set of Autonomous Entities.	Autonomous Agent		Yes	
is conferred by	**definition:** That which confers the social construct or brings it into being. **editorial note:** This may be a body of law or it may be explicitly conferred by one party upon another by virtue of a contract between those parties.	UNION OF Law AND Contract AND Constitution		Yes	

10.3.4.4 Ontology: Statute

URI

http://www.omg.org/spec/FIBO/Foundation/Legal/Statute/

Imports

- LegalCore

Classes

Class: Statute Law

definition: Any law or body of law, passed by some competent authority within some jurisdiction and recognized and enforced in that jurisdiction.

Parents

- Law

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
is in force in	**definition:** The Jurisdiction in which the Statute Law has effect. **editorial note:** This is not the same thing as a Country, but frequently corresponds to one. The Jurisdiction may also correspond to part of a country (England-and-Wales) or to a State or Province within a Federal country, or it may correspond to the federal country itself if the statute is a federal law.	Jurisdiction		Yes	

10.3.4.5 Ontology: LegalCapacity

URI

http://www.omg.org/spec/FIBO/Foundation/Legal/LegalCapacity/

Imports

- LegalConstructs

Classes

Class: Contractual Capability

definition: The capacity to enter into legally binding contracts.
editorial note: This is the capacity which defines Contractually Capable Entity (sometimes labeled as 'Legal Entity') as distinct from 'Legal Person'. In the latter case the liabilities incurred in the contract accrue also to the Legal Person. In the case of contractual capability, the entity has the authority to enter into contracts, whether or not the liabilities accrue to that same entity (which they do if it is also a Legal Person). For Legal Entities which are not Legal Persons, the liability unwinds to some legal person within the structure of the entity, for example a General Partner or a Trustee.

Parents

- Legal Capacity

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
is contractual capability of	**editorial note:** This is a context-specific or relative entity, defined in terms of its being an entity and having this capacity. **definition:** An entity which has this Contractual Capability.	Contractually Capable Entity	is capacity of	Yes	

Class: Delegated Legal Authority

definition: Some capacity vested in some party to give them legal control of some entity, that is, a capacity to undertake legally binding commitments on the part of that entity.
editorial note: All these controls are delegated by some kinds of contracts.

Parents

- Legal Capacity

Class: Legal Capacity

isArchetype: true

Parents

- Legal Construct

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
is conferred by	**definition:** That which confers the legal capacity. This may be a body of law or it may be explicitly conferred by one party upon another by virtue of a contract between those parties.	UNION OF Law AND Contract AND Constitution	is conferred by	Yes	
is capacity of	**editorial note:** This includes capacities specific to duties at law (such as those for corporate officers) as well as the ability or capacity to incur liability itself. **definition:** That upon which the legal capacity has been conferred.	Autonomous Agent	is conferred on	Yes	

Class: Liability Capacity

definition: The ability to be sued at law.
editorial note: Note that for the purposes of this model, this is distinct from culpability (the ability to commit criminal acts). That would be a separate and analogous term but with grounding in criminal rather than civil law.

Parents

- Legal Capacity

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
accrues to	**editorial note:** This is by definition a Legal Entity in the sense used here, that is to say a legal entity is defined by the fact that it is the entity to which any and all liability ultimately unwinds, regardless of whether agreements have been entered into by some other entity or on behalf of some other person. **definition:** That to which the liability ultimately unwinds, or to which it accrues.	Legal Person	is capacity of	Yes	

Class: Signatory Capacity

definition: The capacity of some natural person to sign agreements on the part of some entity.

Parents

- Legal Capacity

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Signatory capacity is capacity of	**definition:** That entity of which the Signatory Capacity is a capacity, i.e. that entity upon which that capacity is incurred. This is always an adult human being (human being which is also a Legal Person).	Natural Person	is capacity of	Yes	

10.3.4.6 Ontology: Contract

URI

http://www.omg.org/spec/FIBO/Foundation/Legal/Contract/

Imports

- LegalCore
- Agreements

Classes

Class: Conditions Precedent

editorial note: Introduced for ISDA Master Agreement. It is likely that the Conditions Precedent defined for OTC Derivatives Master Agreements are actually applicable more widely. However, they are defined within the ISDA terms for now. Modeling note / review question: Modeled as a kind of Terms Set, combining terms and conditions. Should consider whether terms and conditions are

distinct (Condition would then be a separate archetype).
definition: Conditions precedent on some obligation. These are conditions which would alter the Obligation as it is otherwise stated.

Parents

- Contract Terms Set

Class: Contract

isArchetype: true

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
embodies	**definition:** A relationship between two parties, embodied in and created by that contract.	Contractual Relationship		Yes	
has non binding	**definition:** Terms which are included in the contract but are not considered binding. A breach of such terms in the future would not be considered to be a breach of the contract.	Non Binding Terms Set	has terms	Yes	
has party	**definition:** A party to the contract, that is one of the signatories. This is a legal entity which is a signatory to or holder of the contract, and which grants or concedes certain rights and obligations as defined in the contract.	Contract Party		Yes	
counterparty	**definition:** The second party to a contract. In the event that the contract identifies either party as being the principal, this is the other party to that contract.	Contract Counterparty	has party	Yes	
principal	**definition:** The main or principal party to the contract, if either party is identified as such.	Contract Principal	has party	Yes	
has terms	**definition:** A set of terms that form part of the contract. These are generally grouped for convenience in defining terms, such as debt repayment terms, and may or may not equate to a formal clause, section, paragraph or other textual construct of the contract.	Contract Terms Set		Yes	
has third	**definition:** A party which is not signatory	Contract	has	Yes	

Name	Annotation	Type	Parent	Multiples	Inverse
party	to the party but has some role in the overall context defined by the contract.	Third Party	party		
principals identification	**definition:** The identification of the principals of the contract.	Contract Party Identification		Yes	
governed by	**definition:** The jurisdiction under which the contract is governed, as agreed by both parties. In a written contract this is generally identified in a clause identified for example as Governing Law. This is the jurisdiction under which any disputes arising from the contract are to be resolved. **editorial note:** As currently modeled this relationship combines two slightly different senses in which a Jurisdiction may be named in some Contract: the jurisdiction under whose laws the contract is deemed to be in force, and the jurisdiction under which the parties agree to submit in the event of any dispute resolution. Scope Note: One thing to tease out is whether "Dispute Resolution" and other forms of "Governing Law" are one and the same thing or not. Dispute Resolution is uncontroversial, the question is whether there are other implications to Governing Law or if it's the same thing. For instance I may undertake to behave as though I were responsible to a particular authority i.e. a particular set of statutes.	Jurisdiction		Yes	
Effective Date		date		No	
Assignability		yes or no		No	

Class: Contract Clause

definition: A set of contractual terms, grouped according to subject, intent or the type of rights and / or obligations to which it refers.

Parents

- Contract Part

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Clause contains Term	**definition:** An individual term contained within the clause.	Contract Term	Written Contract has part	Yes	

Class: Contract Counterparty

Parents

- Contract Party

Mutually Exclusive Classes

Contract Principal

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Contract Counterparty identity	**definition:** The legal entity which is identified as the counterparty to the contract, and which would be entitled to any rights granted by the principal to the contract.	Contractually Capable Entity	Contract Party identity	Yes	

Class: Contract Part

definition: A formal part of a contract such as a clause or term.

Parents

- Contractual Element
- Written Information

Class: Contract Party

definition: A Party to some Contract.

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Contract Party identity	**editorial note:** It is possible for a party to a contract to be a minor, and in so doing takes the rights but not the obligations to the contract. **definition:** The legal entity which is	Contractually Capable Entity		Yes	

Name	Annotation	Type	Parent	Multiples	Inverse
	identified as a party to the contract.				

Class: Contract Party Identification

definition: The identification of parties to a contract. This is usually but not necessarily identified in the preamble to a contract.

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
identifies contract party as	**definition:** The business entities identified in the contact party identification.	Contractually Capable Entity		Yes	
identifies	**definition:** A party identified in the contract party identification.	Contract Party		Yes	

Class: Contract Preamble

definition: The part of a contract which defines the parties and gives sufficient information for them to be unambiguously identified, along with other pertinent information such as the subject of the contract.

Parents

- Contract Part

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
defines principals	**definition:** The identification of parties to the contract as embodied in the preamble.	Contract Principals Identification		Yes	

Class: Contract Principal

definition: The party identified as being the principal or first party to a contract, in the event that the contract distinguishes either party as being the principal.

Parents

- Contract Party

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Contract Principal identity	**definition:** The legal entity which is identified as the principal of the contract. **editorial note:** This is usually identified as such in the formal preamble to the contract if there is one. It may also be the issuer of a security.	Contractually Capable Entity	Contract Party identity	Yes	

Class: Contract Principals Identification

definition: The identification of the principals to a contract.

Parents

- Contract Party Identification

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
identifies principal	**definition:** The principal to the contract, identified in the contract principals identification.	Contract Principal	identifies	Yes	
identifies counterparty	**definition:** The counterparty to the contract, identified in the contract principals identification.	Contract Counterparty	identifies	Yes	
identifies contract principal as	**definition:** The business entities which are identified as the principals to the contract.	Contractually Capable Entity	identifies contract party as	Yes	

Class: Contract Section

definition: A formally identified Section of a Contract, comtaining terms dealing with a specific type of subject matter.

Parents

- Contract Part

Properties

Name	Annotation	Type	Parent	Multiples	Inverse

Name	Annotation	Type	Parent	Multiples	Inverse
Section contains Clause	**definition:** A clause contained in the contract section.	Contract Clause	Written Contract has part	Yes	

Class: Contract Term

definition: An individual term in a Contract. Forms part of a set of Contractual Terms. Also exists within a Clause of a Contract A Clause exists within a Section of a Contract

Parents

- Contract Part

Class: Contract Terms Set

isArchetype: true

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
has term	**definition:** An individual term within the contractual terms set.	Contract Term		Yes	
has part	**definition:** A set of terms which forms part of the contract terms set. This may be a sub-clause, a clause or simply another definition of a set of terms at a smaller granularity than the set from which these are referred.	Contract Terms Set		Yes	
defined in Clause	**definition:** The individual clause of the contract in which the set of terms is defined, if applicable.	Contract Clause	defined in Section	Yes	
defined in Section	**definition:** The section of the contract in which the contract terms set is defined.	Contract Section		Yes	

Class: Contract Third Party

definition: A party which is identified in a contract but which is not itself party to that contract.

Class: Contract Third Party Identification

definition: The identification of any third parties referred to in a contract but not being signatories to that contract. This would for example be the identification of calculation agents in a derivatives contract.

Parents

- Contract Party Identification

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
identifies third party	**definition:** A third party identified in the third party identification.	Contract Third Party	identifies	Yes	
identifies as	**definition:** The business entity which is identified as a third party.	Formal Organization	identifies contract party as	Yes	

Class: Contractual Definition

definition: The definition of something in some contract or other legal instrument.
editorial note: These are agreed definitions which are then referred to in terms in contracts or other legal instruments.

Parents

- Contractual Element

Class: Contractual Element

definition: Anything which relates to contracts.

Class: Contractual Relationship

definition: A relationship in which two or more parties have some contractual obligations or extend some rights under a contract, to one another.

Class: Non Binding Terms Set

definition: Terms which do not have binding legal standing on the Issuer or Holder.

Parents

- Contract Terms Set

Class: Promissory Note

definition: A promise by the issuer of the note, of some good or benefit to the holder.
editorial note: Unlike a contract, a Promissory Note does not need to be signed by both parties. It is essentially a promise from one party to the holder, of some good or benefit. Promissory notes would

generally by fully fungible. These are modeled as a kind of contract but are essentially a kind of unilateral contract between the issuer and the holder, and some authorities might not see this as a contract at all. Cash is a kind of promissory note, with the issuer being a central bank.

Parents

- Contract

Class: Representations Section

editorial note: This is the contract section which formalizes these aspects of the Agreement between the parties. Things still to tease out: there is a difference at law between a warranty and a representation. There are differences between the implications if one or other proves to be untrue: one will render the contract void; the other renders only specific elements of the contract void or makes some difference between those elements. A warranty is likely a stronger assertion thatn a representation (has greater impact).
definition: Section containing statements held out by one party to the other as being true and correct at the time of the Agreement. A representation, as contained in this section of a contract, is a statement by one other party asserting that some given state of the world exists.

Parents

- Contract Part

Class: Termination Provisions

definition: Formal terms setting out how the written contract may be terminated and what happens when it is.

Parents

- Contract Terms Set

Class: Transferable Contract Holder

editorial note: This party may transfer the contract to another party without reference to the issuer, for example by selling it in some marketplace.
definition: The party which holds a transferable contract and enjoys the benefits defined in that contract while they hold it.

Parents

- Owner
- Contract Counterparty

Class: Warranties Section

editorial note: This is the contract section which formalizes these aspects of the Agreement between the parties. Things still to tease out: there is a difference at law between a warranty and a representation. There are differences between the implications if one or other proves to be untrue: one will render the contract void; the other renders only specific elements of the contract void or makes some difference between those elements. A warranty is likely a stronger assertion thatn a representation (has greater impact).
definition: Section defining what is warranted by either party to the other. A warranty, as contained in this section of a contract, is a statement by one other party asserting that some given state of the world exists.

Parents

- Contract Part

Class: Bilateral Contract

definition: A contract between two specific named parties. The rights and obligations pertaining to either party cannot be transferred to another party without prior written permission or a change to the contract itself.

Parents

- Contract

Class: DRIP

editorial note: This term is awaitilng further analysis and will then be moved to the appropriate financial instrument section.
definition: A kind of financial contract.

Parents

- Bilateral Contract

Class: Transferable Contract

Parents

- Written Contract

Mutually Exclusive Classes

Bilateral Contract

Properties

Name	Annotation	Type	Parent	Multiples	Inverse

Name	Annotation	Type	Parent	Multiples	Inverse
held by	**definition:** The anonymous counterparty that holds the contract and may transfer the rights and obligations therein to another such holder.	Transferable Contract Holder	counterparty	Yes	
written by	**definition:** The party by whom the transferable contract has been written.	Writing Party	principal	Yes	

Class: Verbal Contract

definition: A contract which exists as a result of some verbal exchange.

Parents

- Contract

Class: Writing Party

definition: The party which originates the transferable contract and acts as the Principal in that contract.

Parents

- Contract Principal

Class: Written Contract

definition: A formal Contract which is written and signed by both parties thereto.

Parents

- Contract

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
defines	**definition:** The definition of some matter as agreed by the parties to the contract and written in some part of that contract.	Contractual Definition		Yes	
has provisions	**definition:** Termination provisions as contained in the Contract. These set out the conditions under which the contract may be terminated and the rights and obligations of each party in the event of such termination.	Termination Provisions	has terms	Yes	
Written	**definition:** A formal part of the contract	Contract Part	has	Yes	

Name	Annotation	Type	Parent	Multiples	Inverse
Contract has part	such as a clause or term.		part		
supersedes	**definition:** The or any earlier contract which this written contract supersedes, whether that earlier contract is written or verbal or implied.	Contract		Yes	

10.3.5 Model Section: Information

10.3.5.1 Ontology: InfoCore

URI

http://www.omg.org/spec/FIBO/Foundation/Info/InfoCore/

Classes

Class: Document

editorial note: Documents are usually in sections and contain a multiplicity of information, generally on a given subject and for a given audience. In particular, documents generall have parts. This term is defined so that parts of documents may be defined. This is a difficult concept to define, but there is some value in distinguishing information of this sort, from memoranda, reports and the like. THis is particularly for the identification of things such as contracts, prospectuses and the like, as distinct from the elements of information which may make up their parts, and allowing for the possibility of these having different status at different times and so on (Draft document versus Formal Document). However, certain of those others (e.g. Report) may also be considered documents in some cases.
definition: An information deliverable created in the course of some business process, and with some lifecycle independent of its use to communicate the information therein.

Parents

- Written Information

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
has part	**definition:** Some kind of written information (including possibly another document) which forms part of the document. Term origin:SR modeling	Written Information		Yes	

Class: Written Information

definition: Some information which is written in some format (physical or electronic) and is or may be communicated in some way.

Parents

- Information

Class: Information

isArchetype: true

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
information flow	**definition:** Information Flow, whereby one piece of information becomes another. The information which the information becomes. Term origin:SR modeling	Information		Yes	
derived from	**editorial note:** This is semantically distinct from what the informtion is 'about'. Here, we refer to something (usualy but not necessarily some other information) which is referred to in the course of arriving at this information. Examples include mathematical terms referred to in order to derive this information, which in turn is actually about something else (the 'is about' thing). **definition:** What the information derives from or refers to in order to be complete.	Thing		Yes	
is about	**definition:** What the information is about	Thing		Yes	

Class: Name

isArchetype: true

Parents

- Information

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Name		text		No	

Class: Structured Name

definition: A name with additional details besides simply being a string of text. This may include multilingual support or structuring of name content (as seen for example in financial securities names).

Parents

- Name

10.3.5.2 Ontology: PublishedInfo

URI

http://www.omg.org/spec/FIBO/Foundation/Info/PublishedInfo/

Imports

- InfoCore

Classes

Class: Web Page

definition: A document published on the World Wide Web and accessible via a Universal Resource Locator.

Parents

- Document

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
URL		uri		No	

Class: Published Information

isArchetype: true

Parents

- Information

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
is about	**definition:** The subject matter of the published information.	Thing	is about	Yes	

Name	Annotation	Type	Parent	Multiples	Inverse
published by	**definition:** The organization which makes this information available.	Publisher		Yes	

Class: Publisher

definition: An organization (such as a publishing house) which makes information. available.
editorial note: This is the role of publisher, whether or not the role is filled by a publishing house. For example it may also be the entity which publishes standard contract terms, e.g. an industry association or an exchange.

Parents

- Enabling Agent

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Publisher identity	**definition:** A publisher of information may be identified as any Business Entity, including for example specialized publishing house, ratings agency, standards body and so on.	Formal Organization	has identity	Yes	
Brand Name		text		No	

Class: Publishing House

definition: Some business entity which specializes in publishing information.

Parents

- Formal Organization

10.3.5.3 Ontology: Standards

URI

http://www.omg.org/spec/FIBO/Foundation/Info/Standards/

Imports

- PublishedInfo

Classes

Class: Industry Association

definition: An organization which has membership drawn from members of some industry and acts to serve the interests of those members.

Parents

- Formal Organization

Class: Standards Body

definition: An organization dedicated exclusively to the propagation of standards.

Parents

- Formal Organization

Class: Standards Setter

definition: An agent which acts to set standards such as message standards, programming standards and so on. This may in principal be any kind of business entity. Note that by virtue of its activities the Standards Setter is also a Publisher, that is it publishes versions of the standards that it sets. However the activity of setting a standard is not itself a direct specialization of the activity of publishing, rather it is a by-product of that activity. This is why there is an apparently redundant parent relationship with Publisher.

Parents

- Publisher

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Standards Setter identity	**editorial note:** This could be any type of Formal Organization. **definition:** The entity which performs the role of Standards Setter.	Formal Organization	Publisher identity	Yes	
Body Acronym		text		No	

Class: Technical Standard

definition: An established norm or requirement about technical systems; a published document formally setting out such established norm.

Parents

- Published Information

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Standard set by	**definition:** The agency by which the standard is set. **editorial note:** This may be one of a range of types of entity; this relationship is defined in terms of some entity standing in the role of Standards Setter.	Standards Setter	published by	Yes	
Acronym		text		No	
Standard Name		text		No	

10.3.6 Model Section: Accounting

10.3.6.1 Ontology: AccountingEquity

URI

http://www.omg.org/spec/FIBO/Foundation/Accounting/AccountingEquity/

Imports

- CurrencyAmount

Classes

Class: Equity

definition: A share or proportion of the capital gains of some venture such as a company, a pool investment (fund) or asset pool.

Parents

- Capital

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Equity represents	**definition:** The amount of money represented by the Equity.	Money Amount	represents	Yes	

Class: Financial Asset

definition: An asset consisting of one or more financial instruments, treated as an asset .

Parents

- Asset

Class: Stockholder Equity

definition: Equity held in an entity by stockholders.

Parents

- Equity

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
represents an interest in Formal Organization		Formal Organization		Yes	

Class: Additional Paid In Capital

editorial note: This is not the official accounting industry definition - need to find that!
definition: Capital making up the equity of the entity but not held in any form of stockholder equity.

Parents

- Capital

Class: Balance Sheet Asset

definition: A balance sheet entry representing some asset of the entity.

Class: Balance Sheet Entry

definition: Some entry on the balance sheet of some business entity.

Class: Balance Sheet Liability

definition: A balance sheet entry representing some asset of the entity.

Class: Capital

definition: An amount of money

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
represents	**definition:** The amount of money which makes up the Capital.	Money Amount		Yes	

Class: Owners Equity

definition: Equity owned in the entity as recorded on the books of that entity.

10.3.6.2 Ontology: CurrencyAmount

URI

http://www.omg.org/spec/FIBO/Foundation/Accounting/CurrencyAmount/

Classes

Class: Country

definition: A self-governing geopolitical entity which is recognized as a country by the United Nations. Scope Note: This class is a surrogate for the Country class in the Foundational ontologies; modelers may elect to relate this to Country (or equivalents e.g. Territory) in other ontologies.

Class: Currency

definition: The unit of measure of cash.

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
is tender in	**definition:** A country in which the currency is exchangeable for goods and services. Commonly referred to also as legal tender, however this definition does not hold literally in some countries e.g. Scotland.	Country		Yes	

Class: Monetary Amount

definition: The measure which is an amount of money specified in monetary units.
editorial note: This is an abstract concept, not to be confused with a sum of money ('Money Amount'). This is a "Referenceable Archetype" meaning that in terms of the modeling convention it is possible to create a Relationship Fact which refers to this, and have it appear "inthe box" (in OWL and ODM UML profile terms: this is the range of an object property but is displayed using the UML construct of Attribute).

Parents

- Monetary Measure

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Amount		decimal		No	
Currency		Currency		No	

Class: Monetary Measure

definition: Some measure of some sum of money.
editorial note: This mayt be a measure expressed in terms of decimal plus currency, or it may be a measure expressed in terms of a percentage amount with reference to some other monetary amount or to some 'Money Amount' (actual amount of money).

Class: Money Amount

definition: A sum of money.
editorial note: This is an actual sum of money, not the measure of a sum of money in monetary units, although it has the same basic properties (decimal number with a currenct unit). Update 14 June 2011: Renamed from "Monetary Amount" to "Money Amount" to make this perhaps clearer. This term here should not be the Referenceable Archetype used to denote monetary amounts as a measure. ACTION: Across the model, all references to "Money Amount" (which was called 'Monetary Amount' when these were entered), so be the abstract quantity "Monetary Amount".

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Denomination		Currency		No	

Class: Percentage Monetary Amount

definition: A measure of some amount of money expressed as a percentage of some other amount, some notional amount or some concrete Money Amount.
editorial note: This will have a relationship to what it is a percentage of. Alternatively and for some applications of this term, there may be an enumerated list of possible things it is a percentage of.

Parents

- Monetary Measure

Properties

Name	Annotation	Type	Parent	Multiples	Inverse
Percentage Amount		percentage		No	

Annex A: Deliverables Included with this RFC

(normative)

The following deliverables are included with this specification as document finance/2013-03-02 through 04.

A.1 XMI Files

Machine readable XMI files containing the full content of this specification, including annotation metadata.

This is finance/2013-03-02.

There are two styles of XMI file:

ODM XMI – representing the content of the model as rendered in the Ontology Definition Metamodel. These are Normative.

UML XMI – representing the UML repository content in serialized form for ingestion into any UML tool, without reference to the ODM specification (this requires that the importing UML repository has some means to recognize the ODM constructs as UML extensions). These are Informative.

A.2 OWL Files

Web Ontology Language (OWL) files in RDF/XML format. These are Normative and Definitive.

This is finance/2013-03-03.

Note that these contain all the content of this specification and are not necessarily appropriate for direct application in semantic technology applications without first carrying out further processing. OWL files adapted specifically for semantic technology applications (see Scope section) are not included with this specification, only complete OWL renditions of the individual ontologies. That is, these are not held as being appropriate for use, without modification or reduction in size and scope, for applications in semantic web reasoning or other semantic technology applications (for example, they may or may not be DL complete, decidable, or reasonable over in a realistic length of time).

A.3 Profiles Convenience Document

This is the Sparx Systems "Enterprise Architect" model file from which the currently used ODM constructs were derived. This is Informative.

This is finance/2013-03-04.

Annex B: Interpreting Model Content

(informative)

B.1 Introduction

Audience for this annex: Business Subject Matter experts

The model content is intended by read and understood by business domain experts with knowledge of business entities and legal concepts. It requires no knowledge of modeling theory, technical modeling languages, technology development or data modeling.

The following knowledge is required to interpret the model content:

- Set theory
- Logic
- Business (commerce, law, finance)

B.2 The Model

What the Model Contains

The model described in this specification contains elements called 'Things', simple facts about those things in the form of textual information, and relationship facts in the form of relationships between one 'Thing' and another. Things, simple facts and relationship facts all have textual information, with as a minimum the definition for the term that they represent, plus additional information on usage, review history, sources of terms and definitions and so forth.

Model Views

The content of the model is rendered in two basic modalities: visual information in the form of diagrams, and textual information in the form of tables. The diagrams are available in varying levels of detail and may have been created to show different sets of terms and relationships across or within sections of the model. The textual information may have been created as web based tabular reports or as spreadsheets. These may contain basic information of term, definition and synonym or they may contain additional information about the types of thing or the types of information to which facts in the model refer. These will usually not show relationships between relationships as such information would be difficult to visualize in the tabular format.

Diagrams and tables reflect the information retained in the underlying model repository directly. For example, if two 'Thing' elements have a relationship between them and they appear on the same diagram, the relationship between them will always appear.

Model Diagrams

Diagrams reflect any set of terms in the model, within or across sections of the content. These may be rendered with varying levels of detail. Diagrams created during reviews of the subject matter will typically contain a greater range of terms than diagrams created for presentation to the wider community of potential users. The levels of detail presented in the diagrams typically include:

- Block diagram: contains only Things and Relationship facts
- Simple diagram: contains Things, Simple Facts and Relationship Facts
- Advanced diagram: as Simple Diagram with the addition of relationships between relationship facts

- Locator diagram: as Advanced Diagram; each 'Thing' and relationship fact has a textual indication of its section location
- UML diagram: as Locator diagram, with UML indications turned on for UML stereotypes and the like. These are not intended for review or consumption by business domain experts and are included for maintenance only.

B.3 Interpretation

The model conveys 'Things' and 'Facts'. Facts are in two forms:

- 'Simple Facts': these are a statement about something which is framed in terms of some simple type of information, such as textual entries, yes/no answers, dates, numbers and selections of textual information
- 'Relationship Facts': these are a statement about something which is framed in terms of something else, that other thing also being framed as a 'Thing'.

In addition, there are relationships which represent additional set theory concepts, notably logical unions, mutual exclusiveness.

Each 'Thing' also has a 'Parent' relationship, with the sense of 'is a'. This relationship indicates that the thing from which that relationship is shown is a kind of the thing to which it refers.

These concepts are described in the sections which follow.

Thing

A Thing is a set theory construct. This is shown on the diagrams as a box, with a textual entry showing its name. On some diagrams, additional textual entries in the box show the simple facts about that thing.

A Thing is defined as the set of individuals which are defined according the facts stated for that kind of thing. Membership of the set is defined in the sense that any individual in the world of which the stated facts are true, is a member of that set. In terms of logical theory, these sets are defined intensionally. It is also possible to define a set explicitly as a list of its members (in logical theoretic terms, an extensional definition) but this is not used in practice in the model.

Inheritance: the Parent 'is a' relationship

Each Thing in the model has one or more parent Things. The relationship between the Thing and its parent may be interpreted as an 'is a' form of relationship, meaning that the thing of which the parent relationship is shown is a kind of the thing to which the arrow in the Parent relationship is pointing.

This relationship is defined according to an Aristotelian syllogism. Aristotle defines four basic syllogisms; the one indicated by this relationship is known as the 'BARBARA' syllogism, and formally indicates that the thing that has the Parent, inherits all the facts about that parent. In addition, this relationship is transitive, meaning that the parent relationships of the parent are passed on to the child term.

The relationships of this type create a formal inheritance tree called a Taxonomy. Taxonomies in this sense may be single inheritance (as is often seen in technical model designs) or multiple inheritance.

As an example of multiple inheritance, one might say that in terms of the Linnaeus Taxonomy of Species, a whale is a mammal, while one may also create a set of taxonomic classifications based on habitat, in terms of which a whale may also be a marine animal.

On a technical note, the Parent relationship is functionally identical to the relationship known as 'Generalization' in the UML modeling language; this is because both languages derive the meaning of this relationship from the above Aristotelian logic. For this reason we have chosen to use the same visual indication for this relationship as it used in the UML language.

This relationship is formally known as 'sub type of' but is labeled in reports as 'Parent'.

Simple Facts

Simple facts are assertions about things in a given class, which may be framed in terms of some simple type of information.

Types of information about which simple facts are asserted are:

- Text
- Date
- Number
- Whole number
- Yes/no answer
- Selection of answers

To a technical person these may easily identified with what are called 'datatypes' however at the level of this model these represent the types of information not data as such. A special case is the selection of possible answers - this refers to a list of entries (see Selection Lists).

Relationship Facts

A relationship fact is defined as a fact about something which is framed in terms of a relationship to some other thing.

These are indicated on the diagrams as a blue arrowed line. Some diagrams additionally show a box attached to this blue line; this is used to indicate relationships between those relationship facts, which are shown as lines between those boxes.

Relationship facts are of the form subject-relationship-object where the subject is the Thing from which the line is drawn and the object is the thing to which the blue arrow points.

The label on the line is the verb itself, while the attached box indicates the full name of the relationship fact. Relationship facts are unique across the model and each belongs to one Thing only.

There are additional pieces of information about these relationship facts, such as whether they are symmetric, transitive and so on. The use and interpretation of these refinements to relationship facts are beyond the scope of this explanatory annex, and these are rarely used in practice in the model to date .

Logical Unions

Logical unions indicate that any individual which is a member of any of the classes of 'Thing' of which the union is a union, are members of that union.

The Union is shown as a box on the diagrams, similar to the boxes used for classes of 'Thing' but without the coloring given for archetypes (no Union has an archetype), that is these have the default gold box appearance of an OWL Class.

Membership of the union is indicated by a purple relationship similar in appearance to the Parent / 'is a' relationship. The Union (set) shown at the top of the arrow is thereby indicated as being a logical union of all the sets indicated as classes of Thing at the bottom of the purple arrows.

Relationship facts may refer to unions in the same way that they refer to other classes of Thing.

Mutually Exclusive sets

Given that each thing is a set of potential members defined by their properties (facts), it is possible for any one thing in the world to be defined as being a member of more than one set, if the properties asserted for one set are not related to the properties asserted for another set.

Where membership of one set necessarily precludes membership of another set (that is, where a set is defined such as to specifically exclude members of another set), this is shown by a red line on the diagrams, labeled 'mutually exclusive'.

Where classes of 'Thing' are not indicated as being mutually exclusive (or have parents which belong to classes of Thing which are mutually exclusive), then any individual in the domain of discourse (the world) may belong to both sets.

This is formally known as a 'disjoint' relationship.

Relationship Facts hierarchies

Relationship facts are themselves disposed in a hierarchy similar to that given for the classes of 'Thing'. These are indicated on more advanced diagrams by a green upward pointing line in the same style as the Parent relationship line. The relationship fact to which the arrow points represents a more general meaning, of which the relationship fact at the bottom of the relationship represents a narrower definition of the same meaning.

The narrowing of these meanings frequently occurs in conjunction with the narrowing of the meanings of classes of 'Thing' in the taxonomy. For example, types of bond are classified (a narrowing or specialization of the meaning of 'bond') according to, among other things, a narrowing of the relationship 'issued by' with the latter relationships being distinguished form one another by the nature of the kind of party which is the issuer.

This is formally known as a #sub property of' relationship.

Inverse relationships

These are only shown on diagrams which show the relationship facts with their boxes, i.e. diagrams which show relationships between relationships.

Relationship facts in the model are all one-directional, by virtue of their being framed as 'subject-verb-object' triples. In the business domain, meaningful terms and definitions may exist in either direction between one class of thing and another (for example, a bank has a customer versus a person has an account at the bank.

These are indicated as a red dotted arrowed line between one relationship and the relationship to which it is the inverse.

In theoretical terms, this relationship only applies between relationships which are known as 'functional' relationships. An explanation of this is beyond the scope of this annex.

Selection Lists

A list of possible entries for a simple type is displayed as a box on the diagrams, with a list of the possible entries. These are displayed as text, and generally refer to lists of possible textual values for the simple fact.

It should be noted that these do not or should not represents lists of kinds of 'Thing' - those would be represented as a taxonomy of actual things. This is an important difference between this and a data model, since many data models have similar selection lists, call 'enumerations' in the data modeling world, which may represent kinds of thing or classifications of the thing which has these as a property.

Selections of Things

This is a class or set of things of which the members are explicitly listed (in theoretical terms, an extensional definition of the class).

These are not used at present in the model but are provided for in the modeling notation.

Annex C: Model Diagrams

(informative)

This Annex is normative for this specification only - terms in the Global Terms section are required for normative use and extension of this ontology but are not held as being normative for other usage outside of this specification.

C.1 Overview

This Annex shows the Business Entity model content. All diagrams are rendered in such a way that they may be viewed full size by zooming in to the page by up to 500%.

Diagrams are included in full, with details of classes of thing (the colored boxes), simple facts, relationships between the classes (the blue lines with associated light blue box), and relationships among those relationships (sub-property relationships shown as green upward arrows between the relationship boxes; red dashed lines showing inverse pairs of relationships). Disjoints (indicating that one class of thing is mutually exclusive in its potential membership with another) are also shown as red dashed lines.

Simpler forms of these diagrams are also available but are not included here. For business review purposes it is recommended to use either block diagrams, or similar diagrams to these but with no display of the relationships among relationships.

Showing diagrams only (where a section has no diagrams, the heading is retained for completeness).

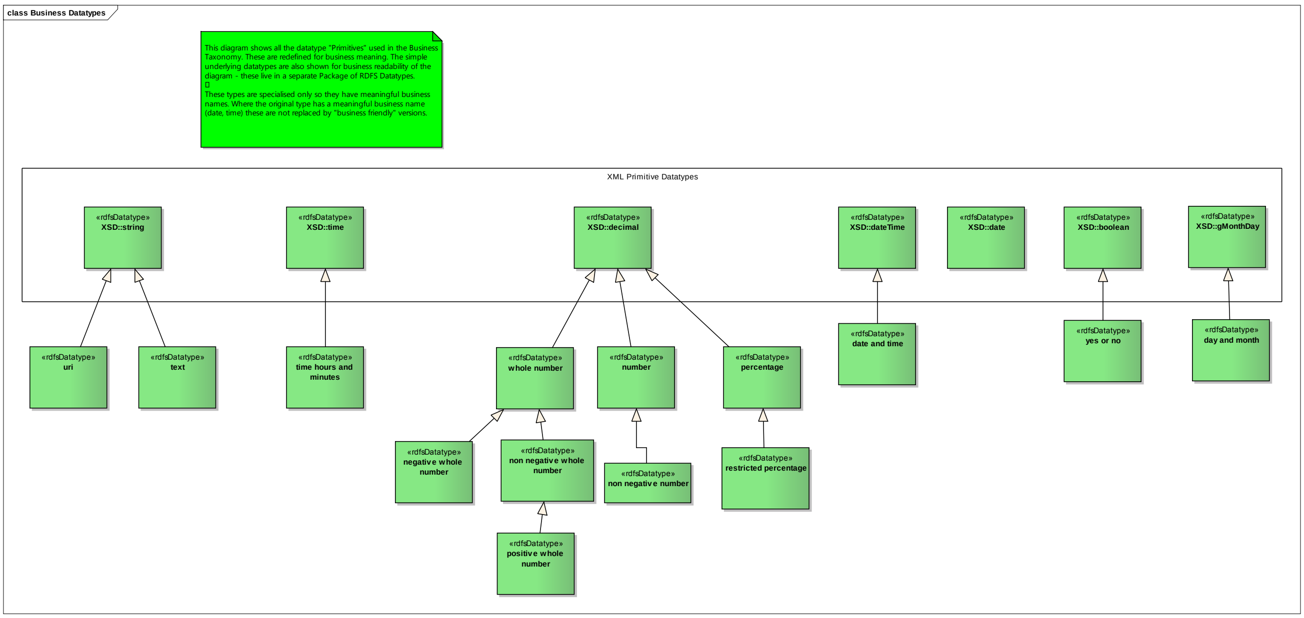
Diagram types:

- Advanced: shows all or most content including relationship hierarchies and inverses
- Simple: shows all except relationship hierarchies and inverses
- Locator: Shows the location of all concepts that are not in the ontology that is reported on (if no location shown, the item is in the section you are looking at)
- Diagram: an extract of the content, created at some time for a more business-facing diagram or presentation

Many of the diagrams contain notes from ongoing discussions. These will not be present in the final versions of these.

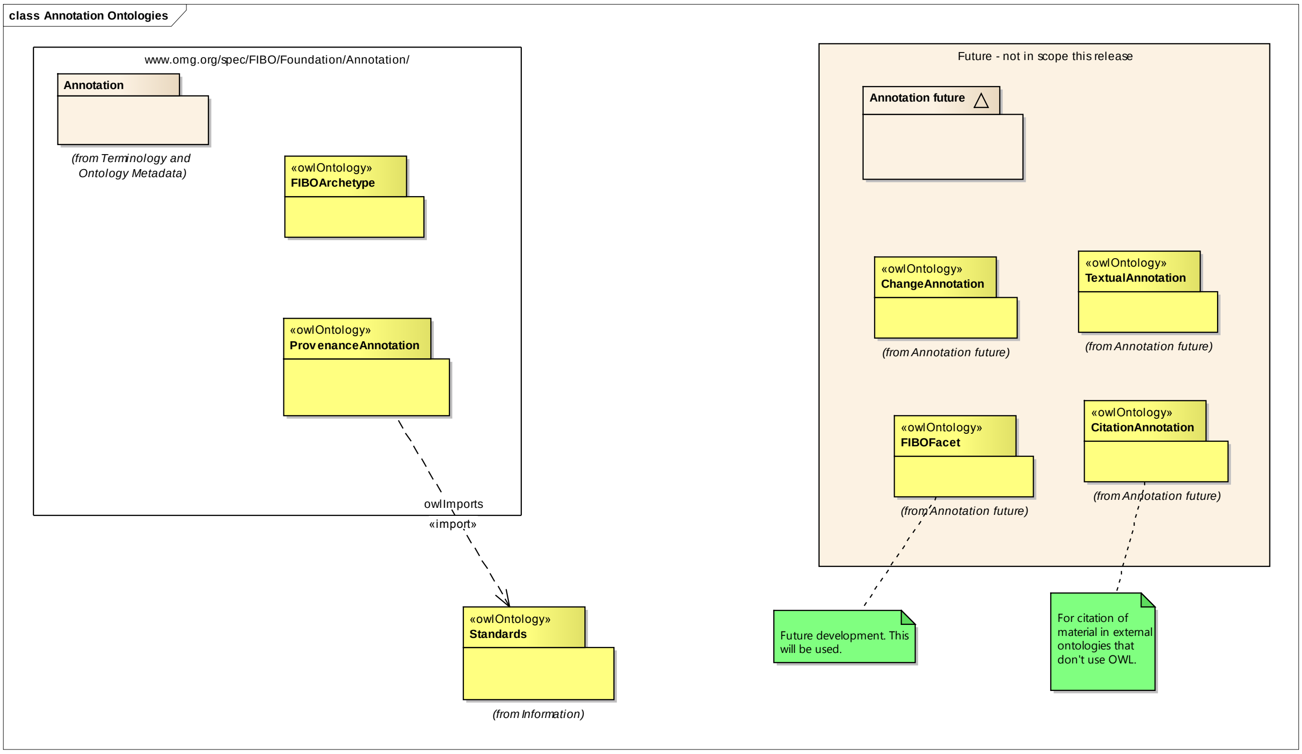
C.2. Ontology Vocabulary

This section contains a copy of the XML Primitive datatypes, along with “Natural Language” extensions or aliasing of those datatypes to enable business-facing presentation of model views which name the type of information that is the target of datatype properties without the use of technical jargon such as “Boolean”. The “Natural Language” datatypes are shown in the figure below. Note that there are also some specific extensions of the XSD “decimal” datatype with specific interpretations such as “percentage”, for business use.



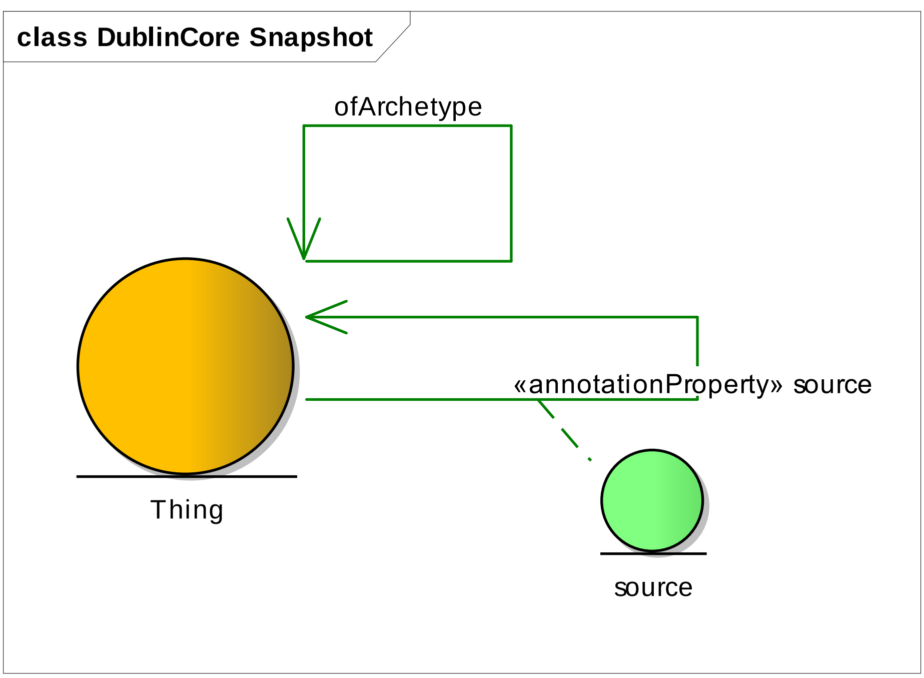
Business datatypes, showing derivation from the XML Primitive (XSD) datatypes

C.3 Annotation



Overview: Annotation Ontologies

C.3.1 DublinCore

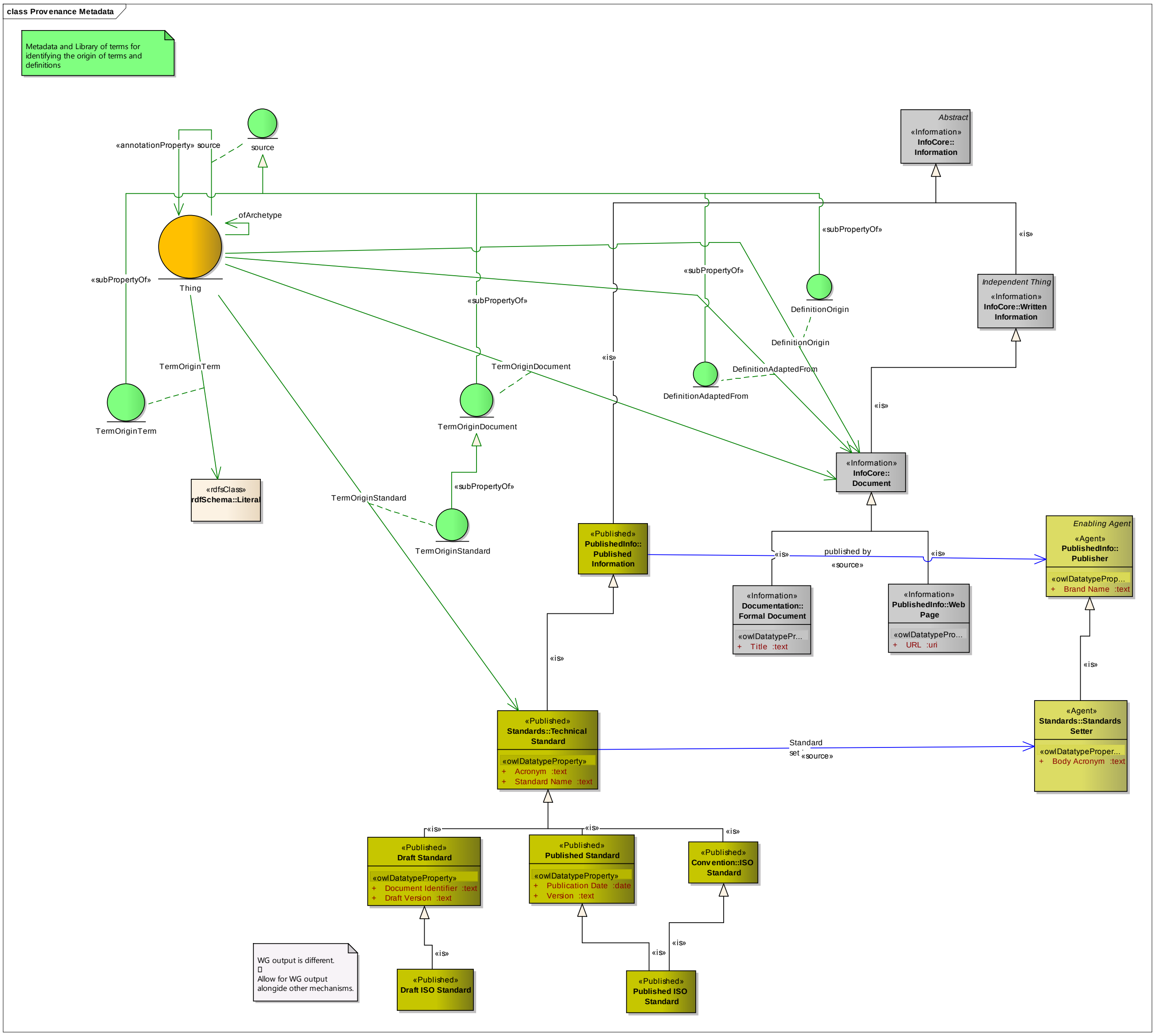


Snapshot of Dublin Core annotation

C.3.2 rdfSchema

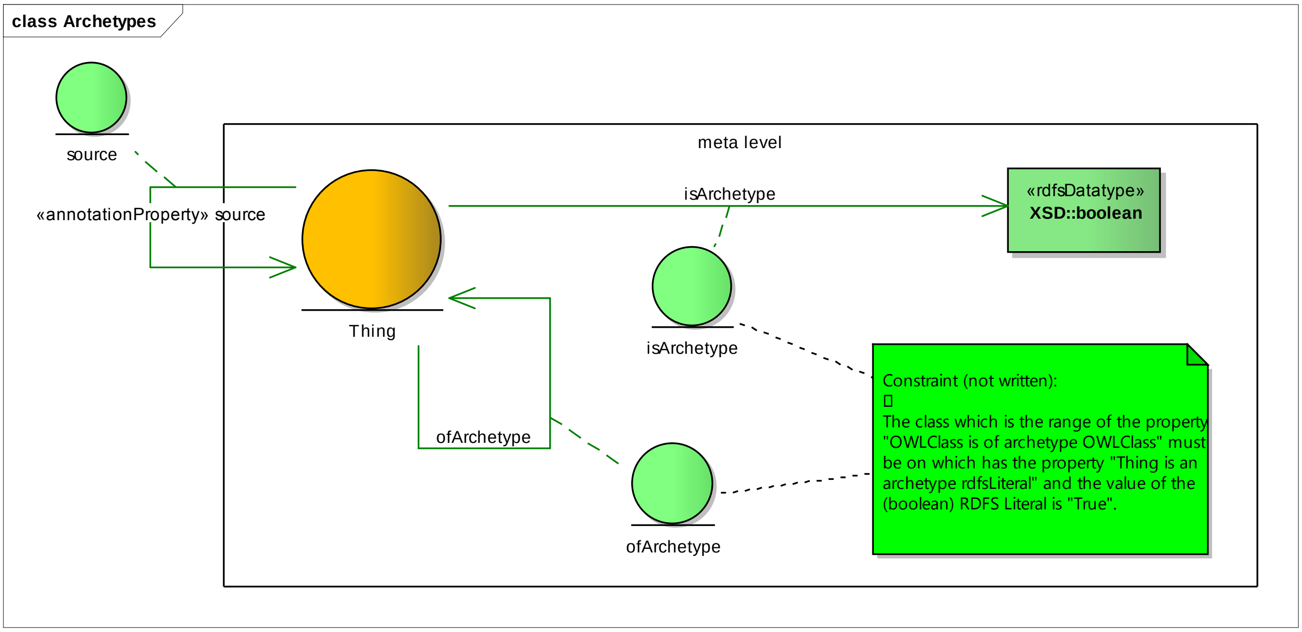
No diagram.

C.3.3 ProvenanceAnnotation



Provenance Annotation Metadata (note that in this rendition, OWL Class and OWL Annotation Property stereotypes are indicated by the circular shapes).

C.3.4 FIBOArchetype



FIBO Archetype annotation properties

C.4. FIBO-Foundation Ontologies

C.4.1 GoalsAndObjectives

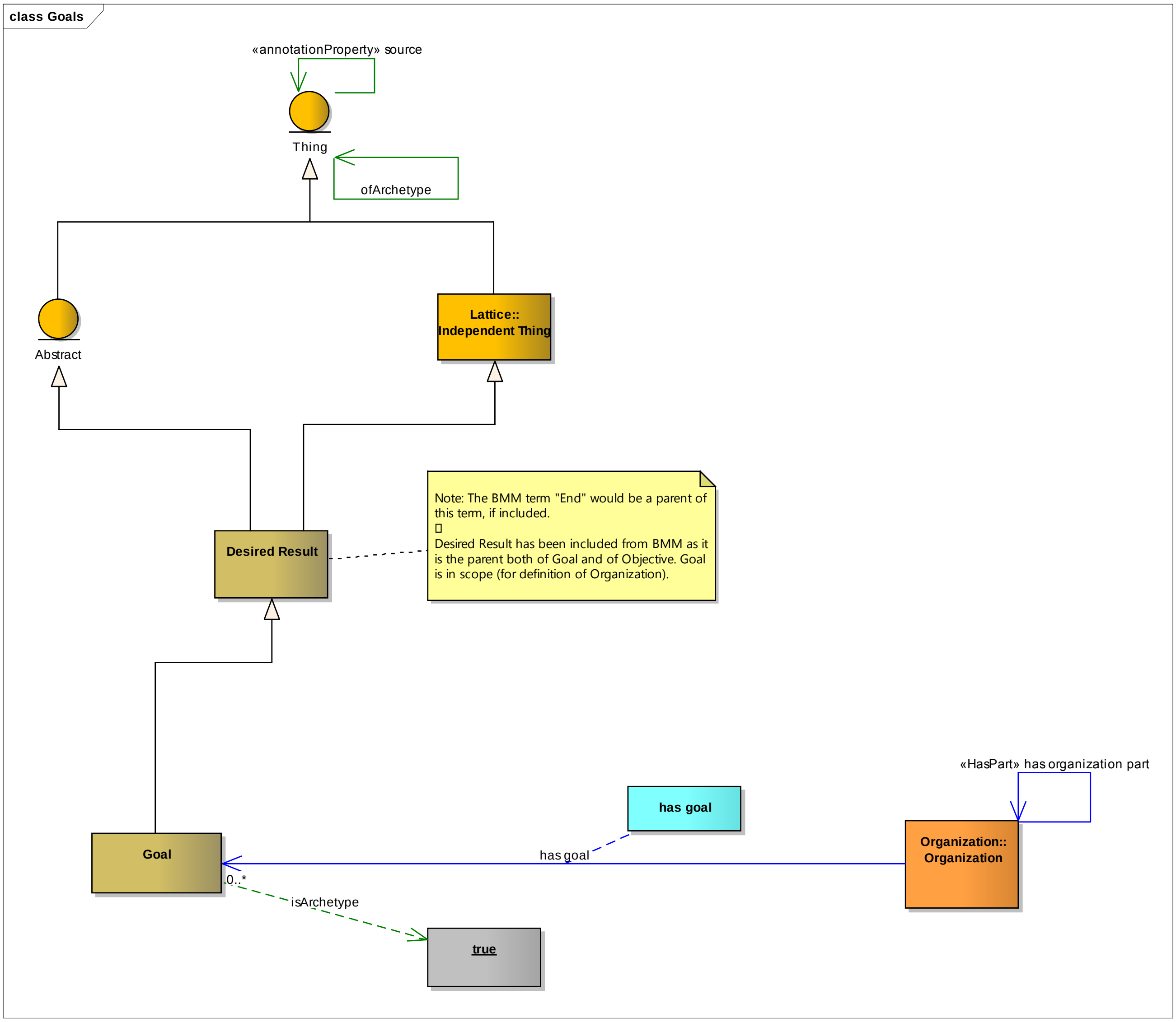
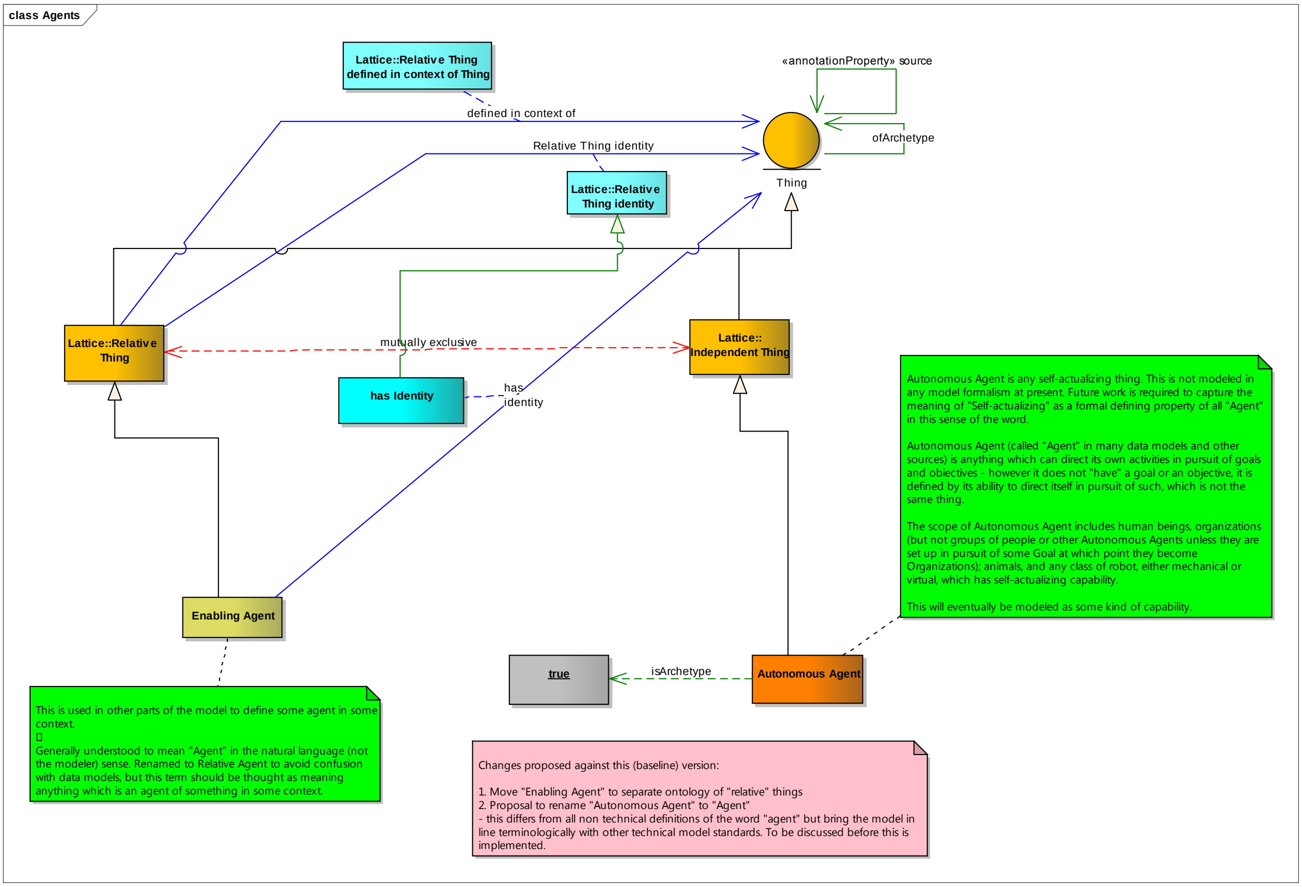


Diagram of Goal and the related term Desired Result.

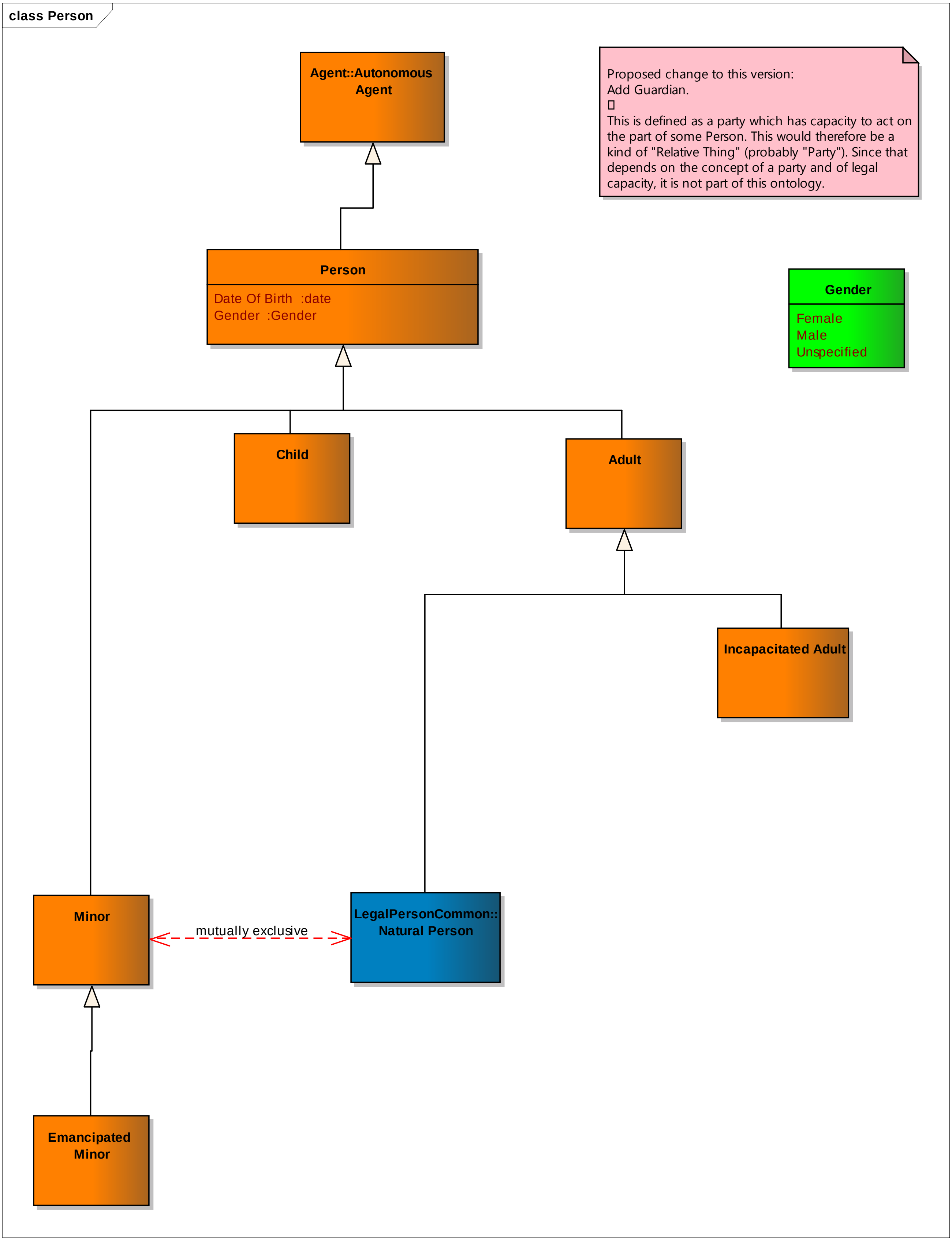
C.4.2 AgentsPeople

C.4.2.1 Agent



The ontology for Autonomous Agent (known as Agent in most technical models); showing change proposals to this version

C.4.2.2 Person

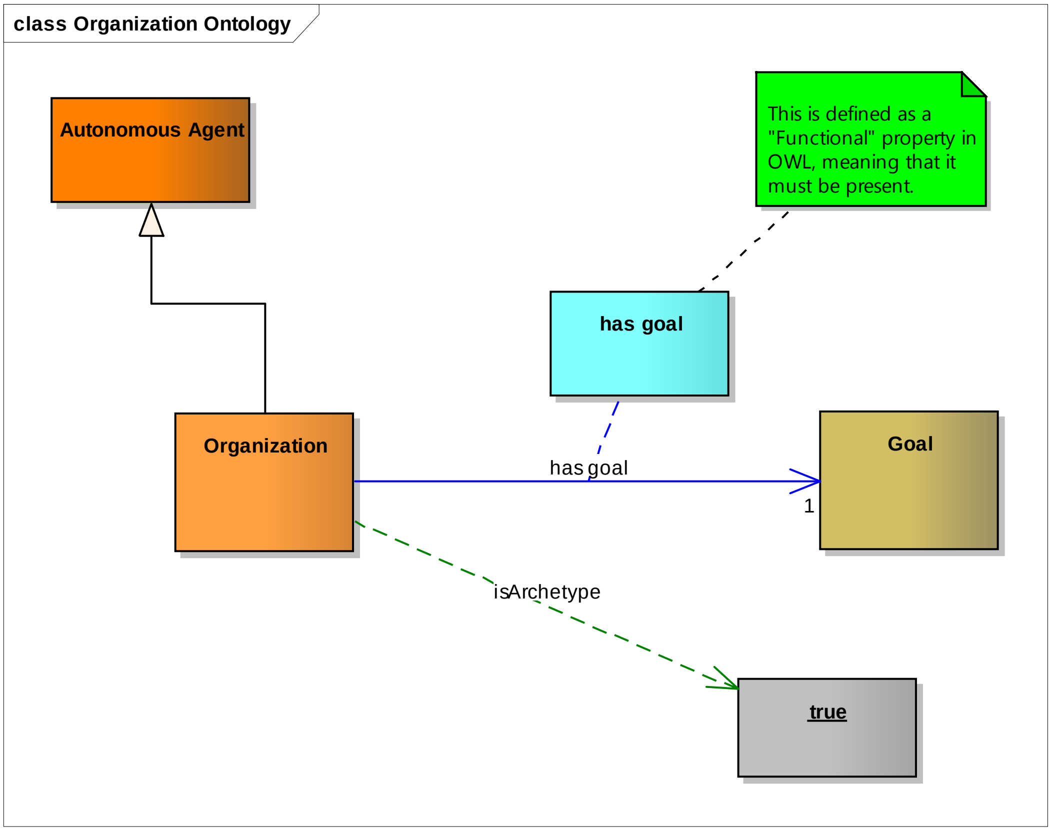


class Person
Agent::Autonomous Agent
Proposed change to this version:
Add Guardian.
This is defined as a party which has capacity to act on the part of some Person. This would therefore be a kind of "Relative Thing" (probably "Party"). Since that depends on the concept of a party and of legal capacity, it is not part of this ontology.
Person
Date Of Birth :date
Gender :Gender
Gender
Female
Male
Unspecified
Child
Adult
Incapacitated Adult
Minor
mutually exclusive
LegalPersonCommon:: Natural Person
Emancipated Minor

The ontology for Person and types of person (other than those in FIBO-BE)

C.4.2.3 Organization

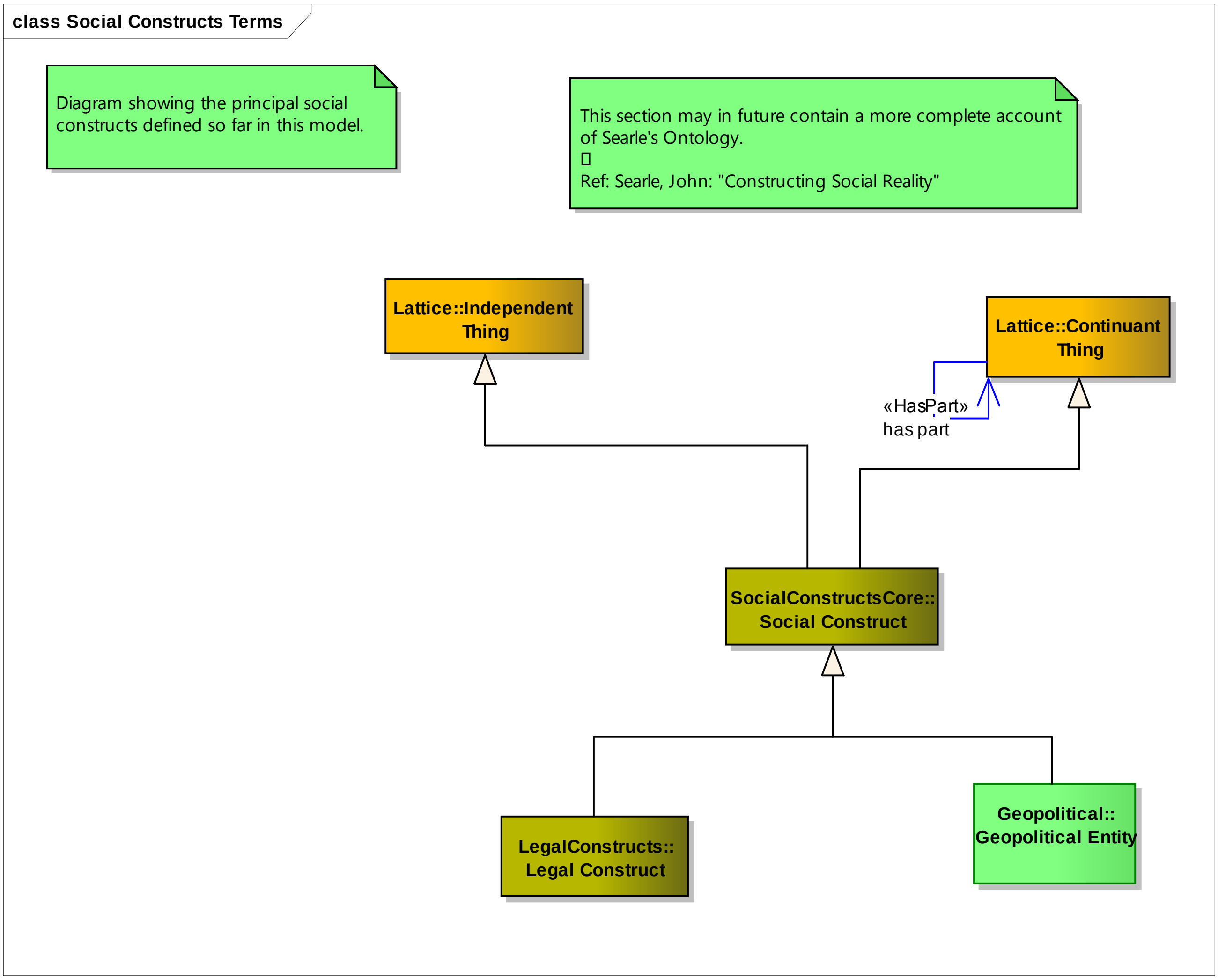
Note that additional properties around organizations (organization types, organization parts and organization members) are included in the FIBO-BE ontology and are not shown here.



Organization Overview

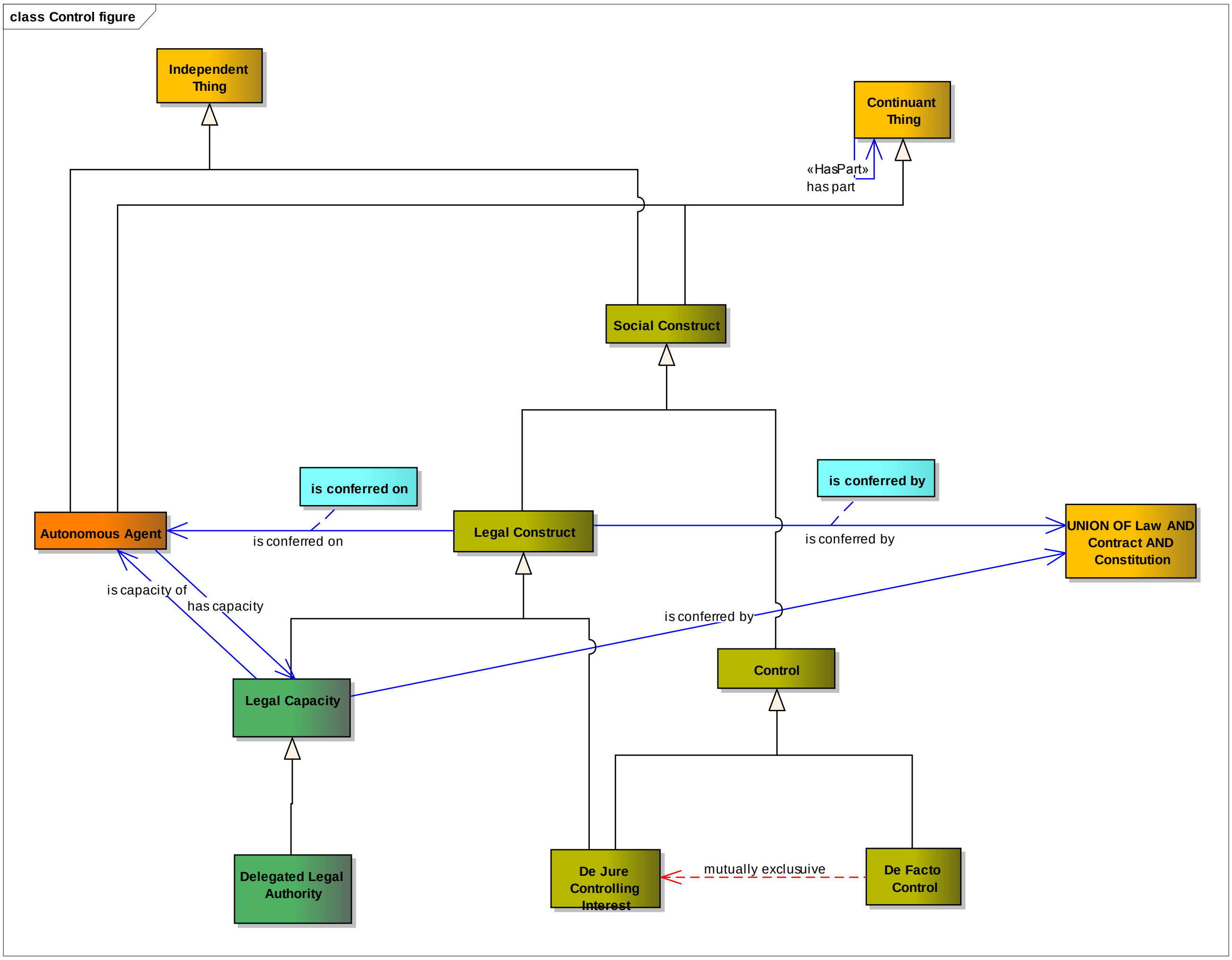
C.4.3 SocialConstruct

C.4.3.1 SocialConstructCore



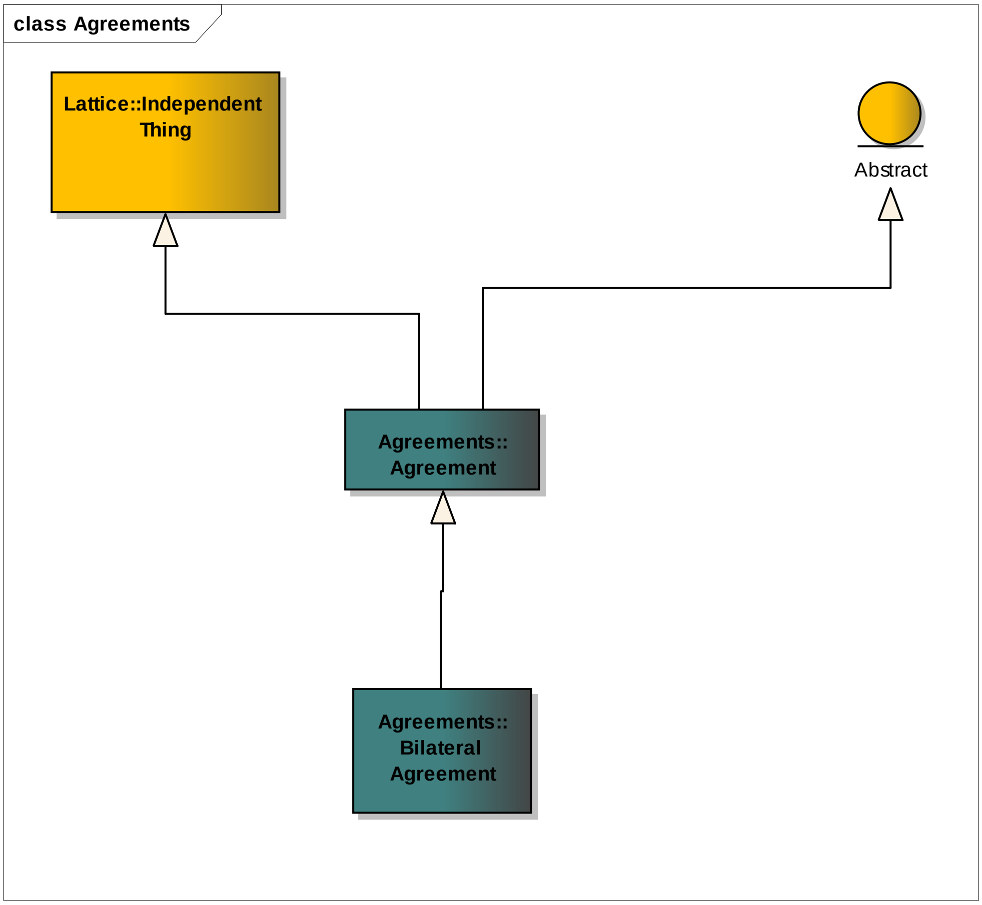
Social Construct core ontology (one term); showing location of elements

C.4.3.2 Control



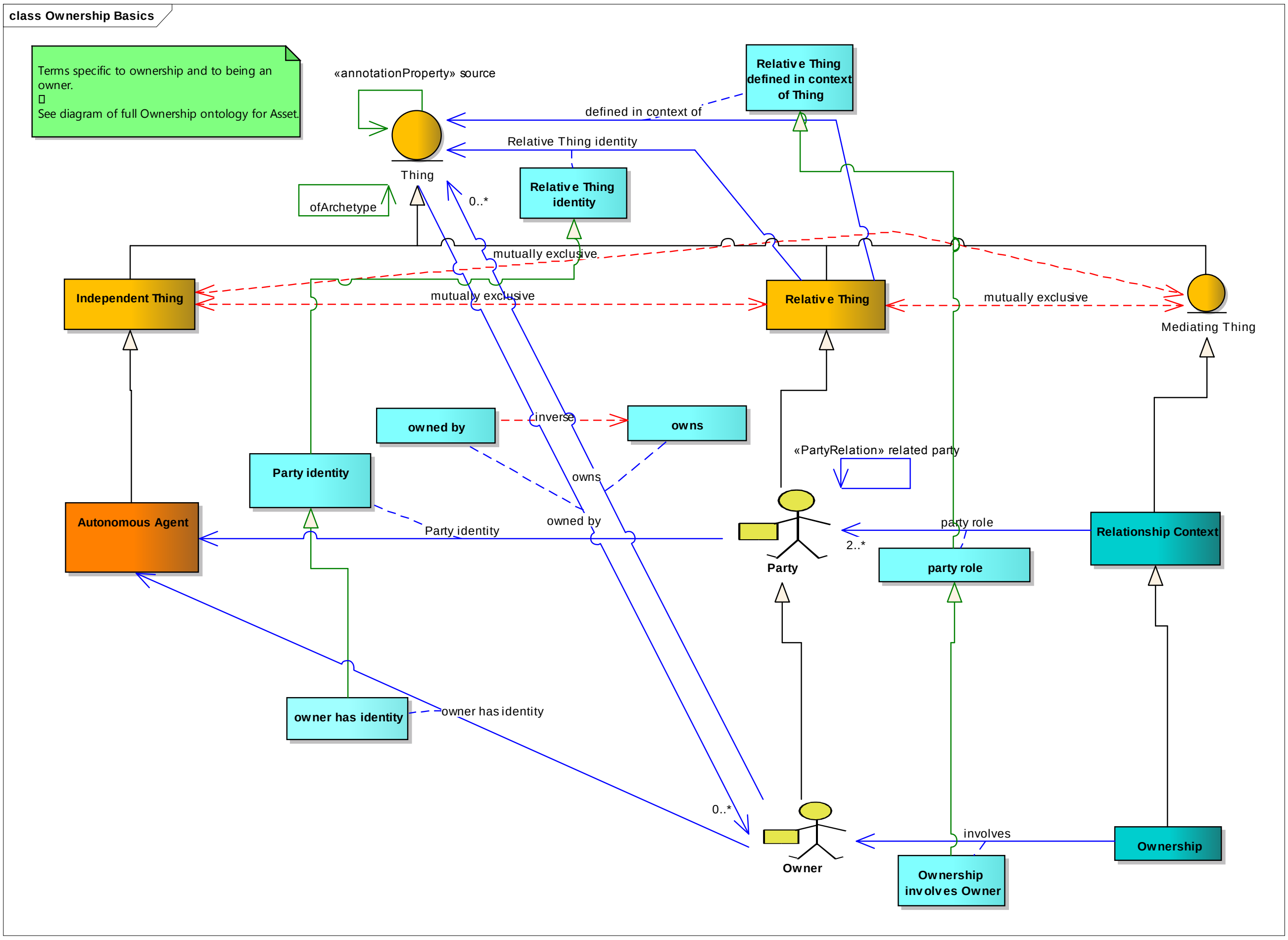
Control Concepts

C.4.3.3 Agreements

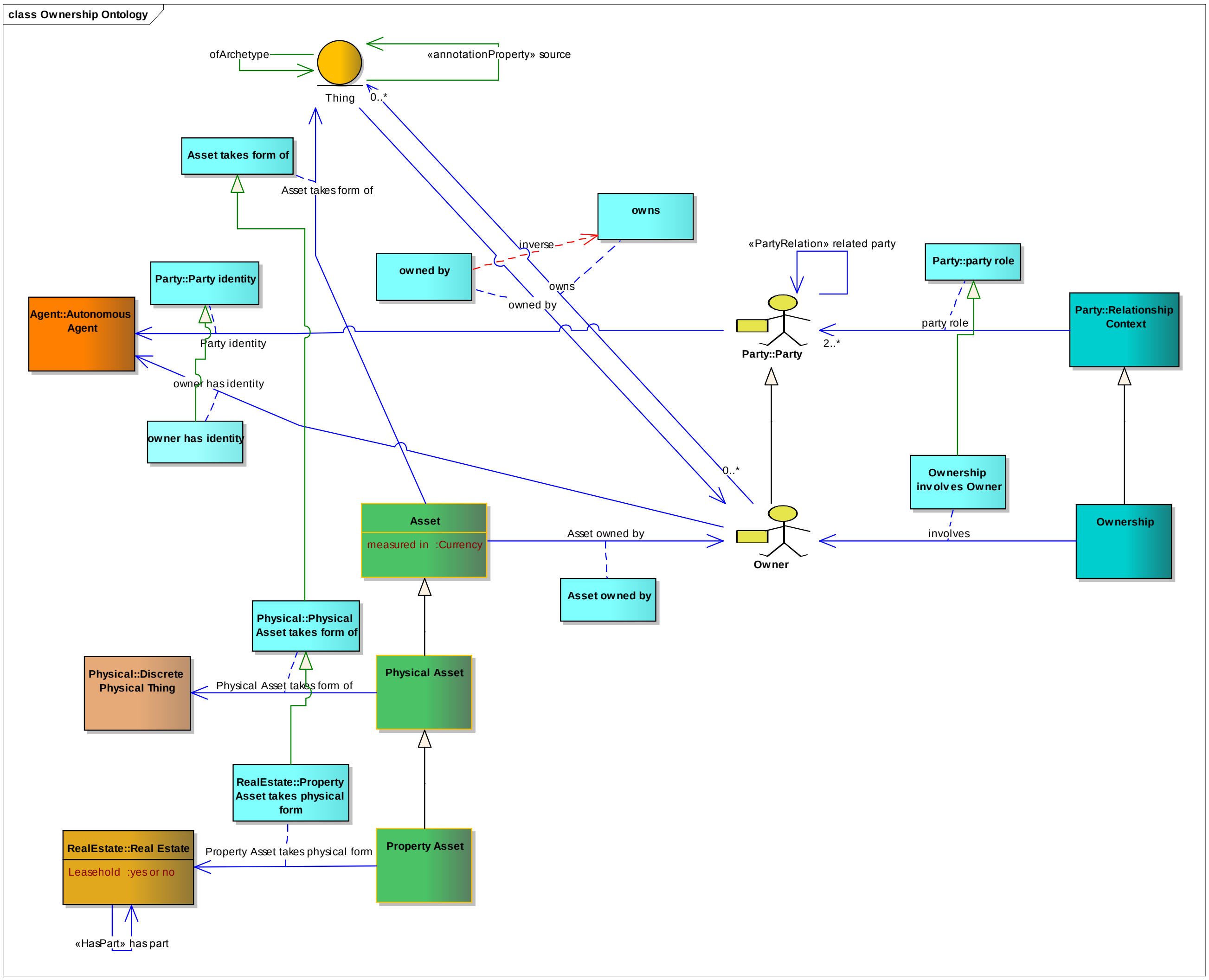


Agreement Terms

C.4.3.4 Ownership



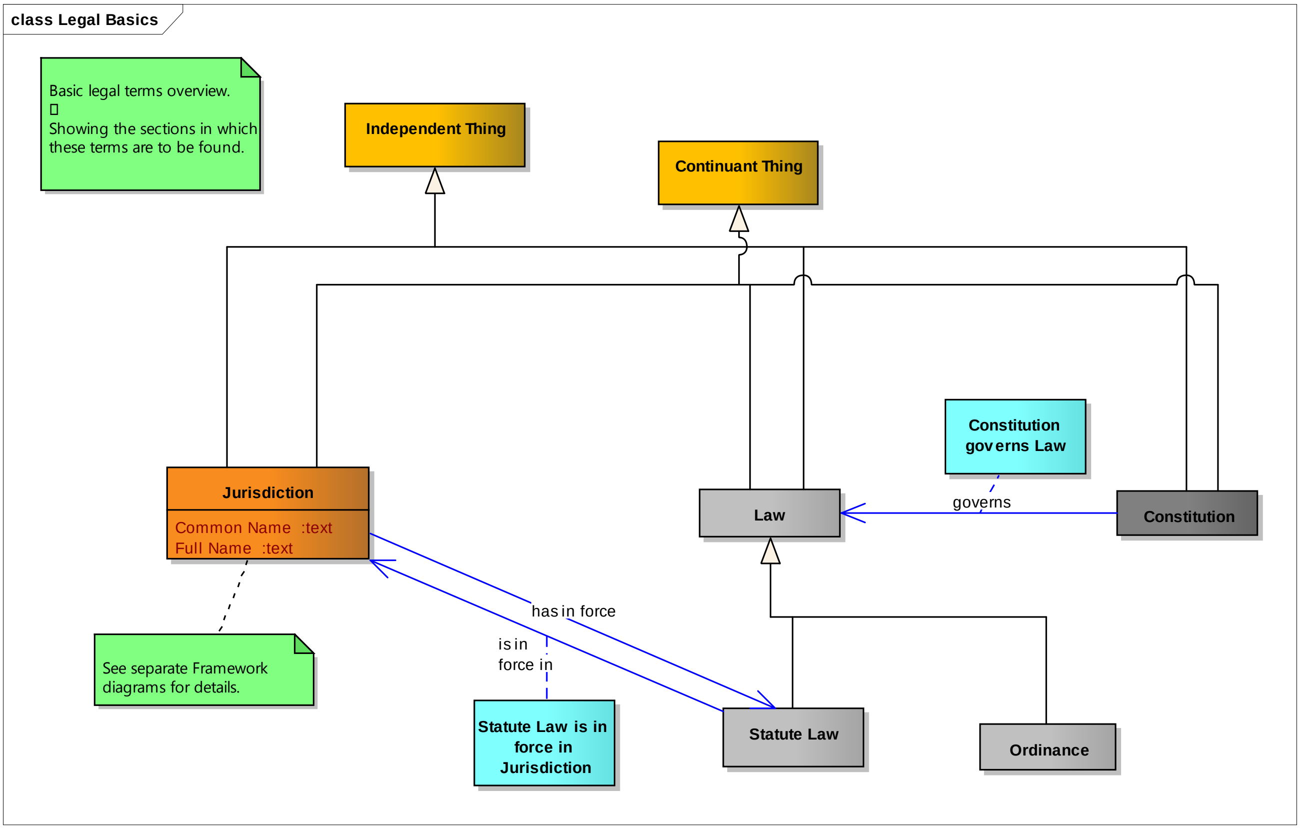
Basic Ownership properties



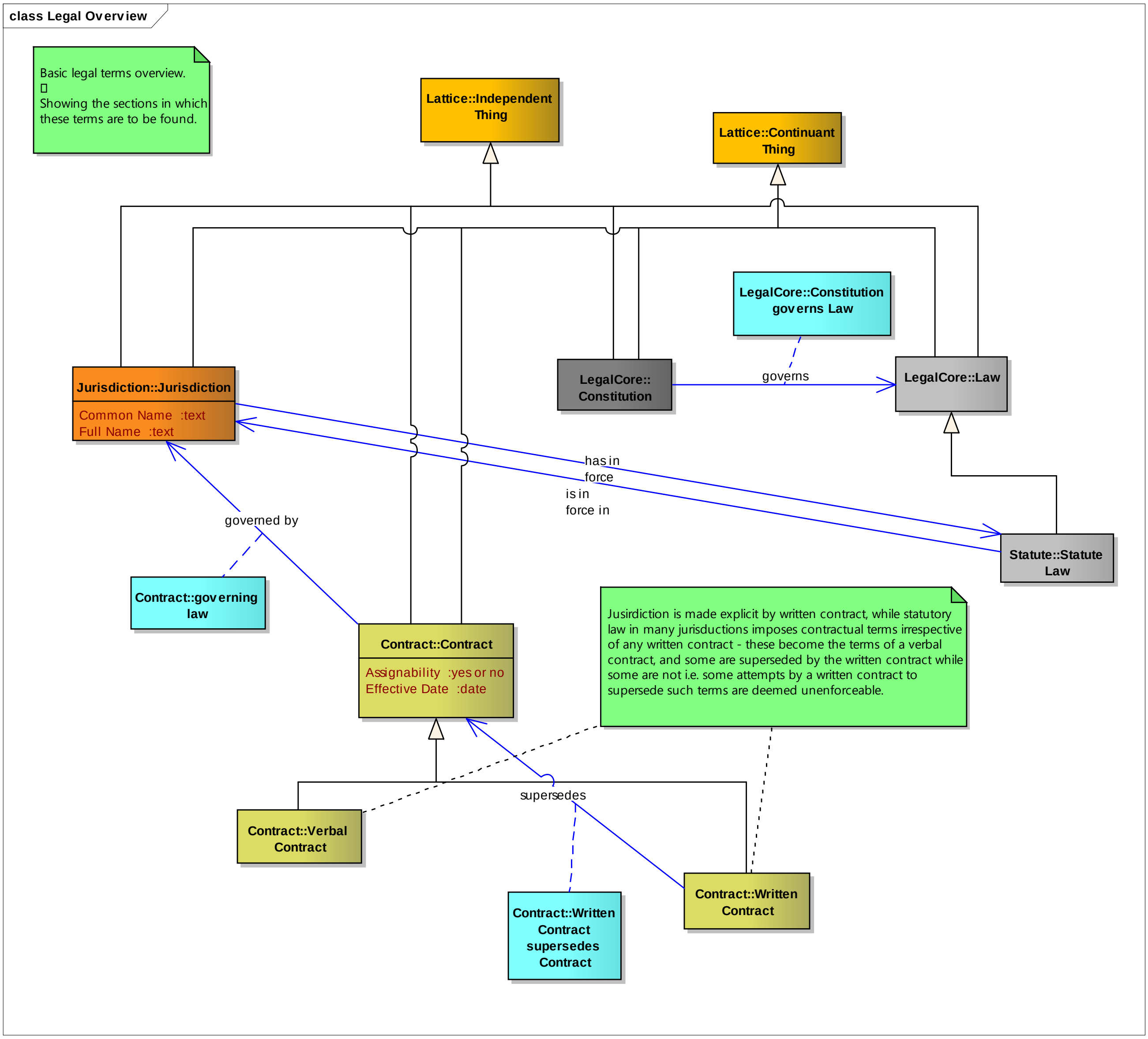
Ownership all concepts in this ontology, along with external terms related to these.

C.4.4 Legal

C.4.4.1 LegalCore

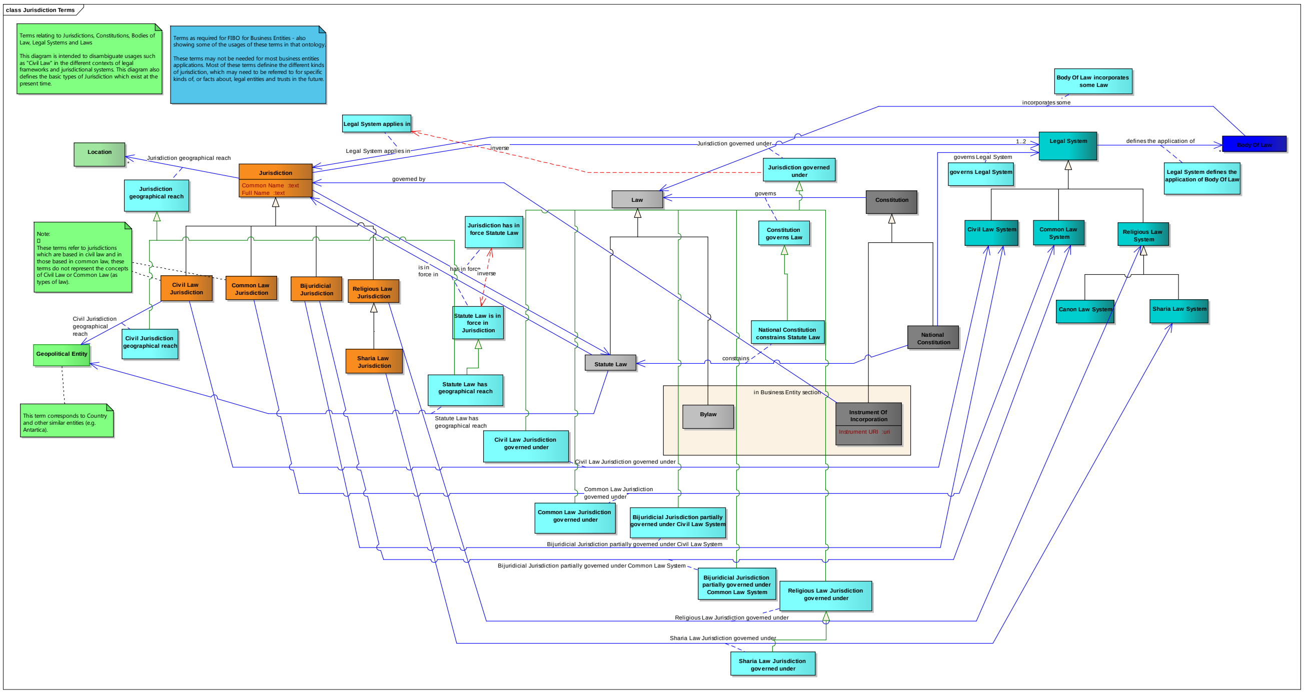


Core legal terms



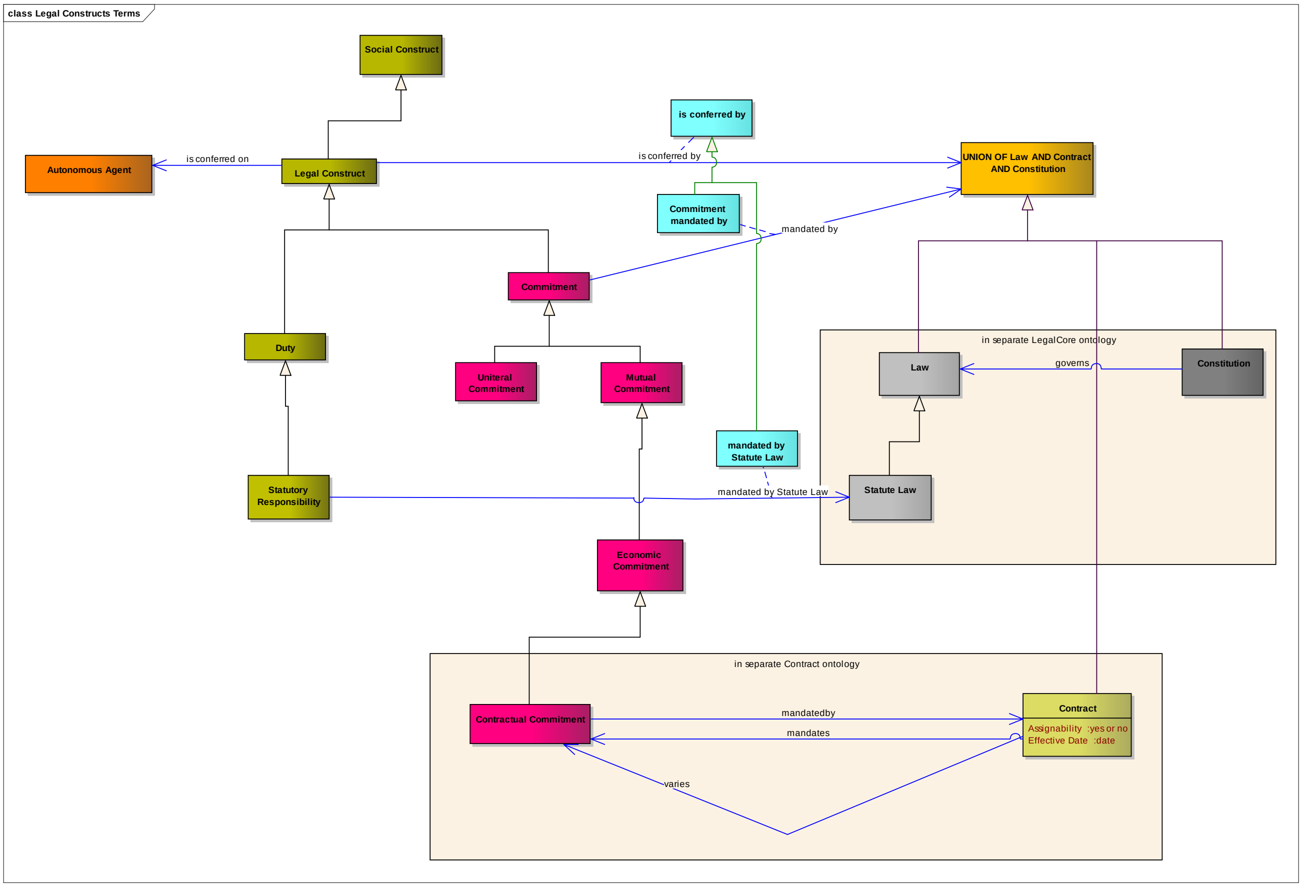
Legal Terms Overview showing location of terms

C.4.4.2 Jurisdiction



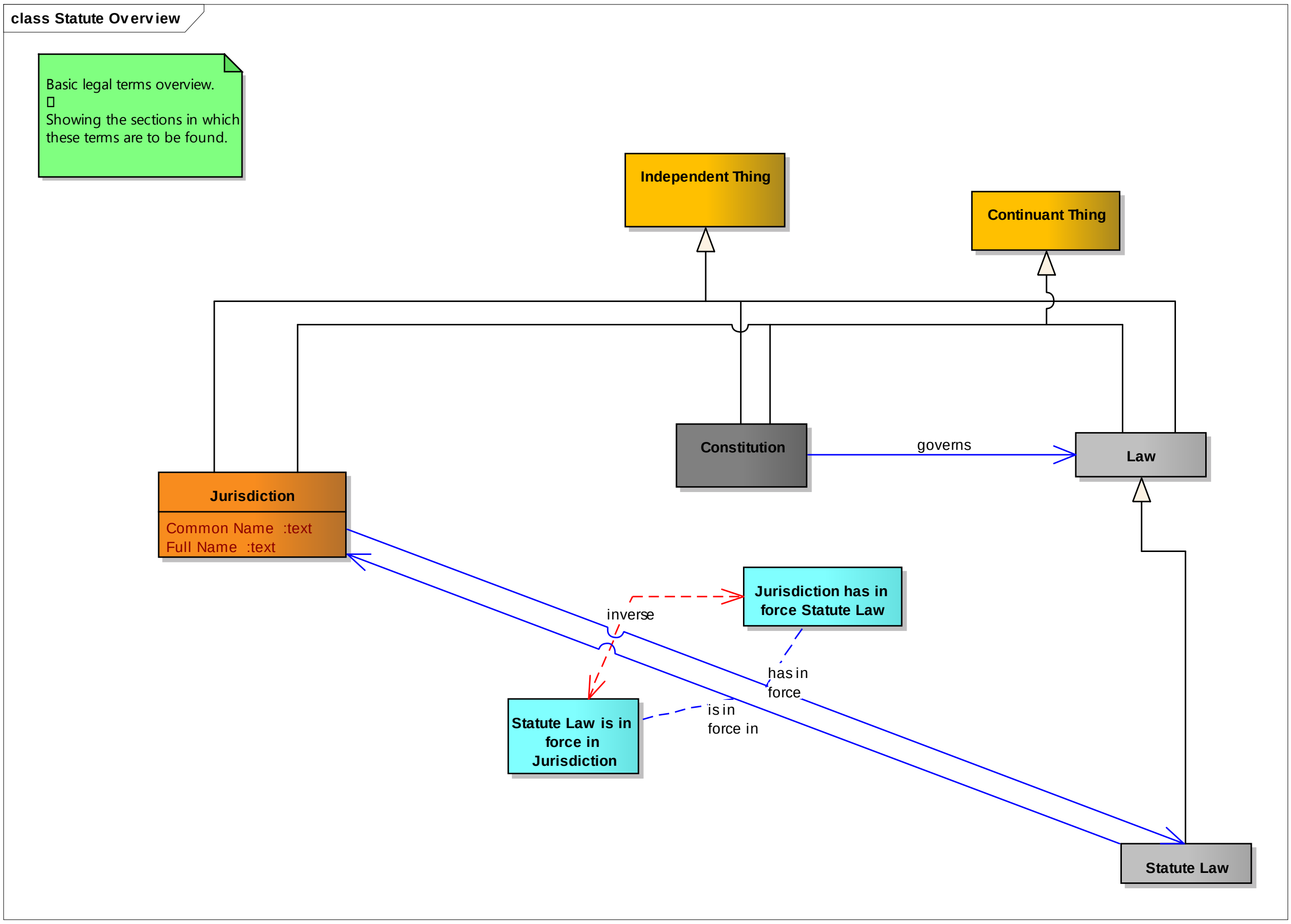
Jurisdiction Concepts

C.4.4.3 LegalConstructs



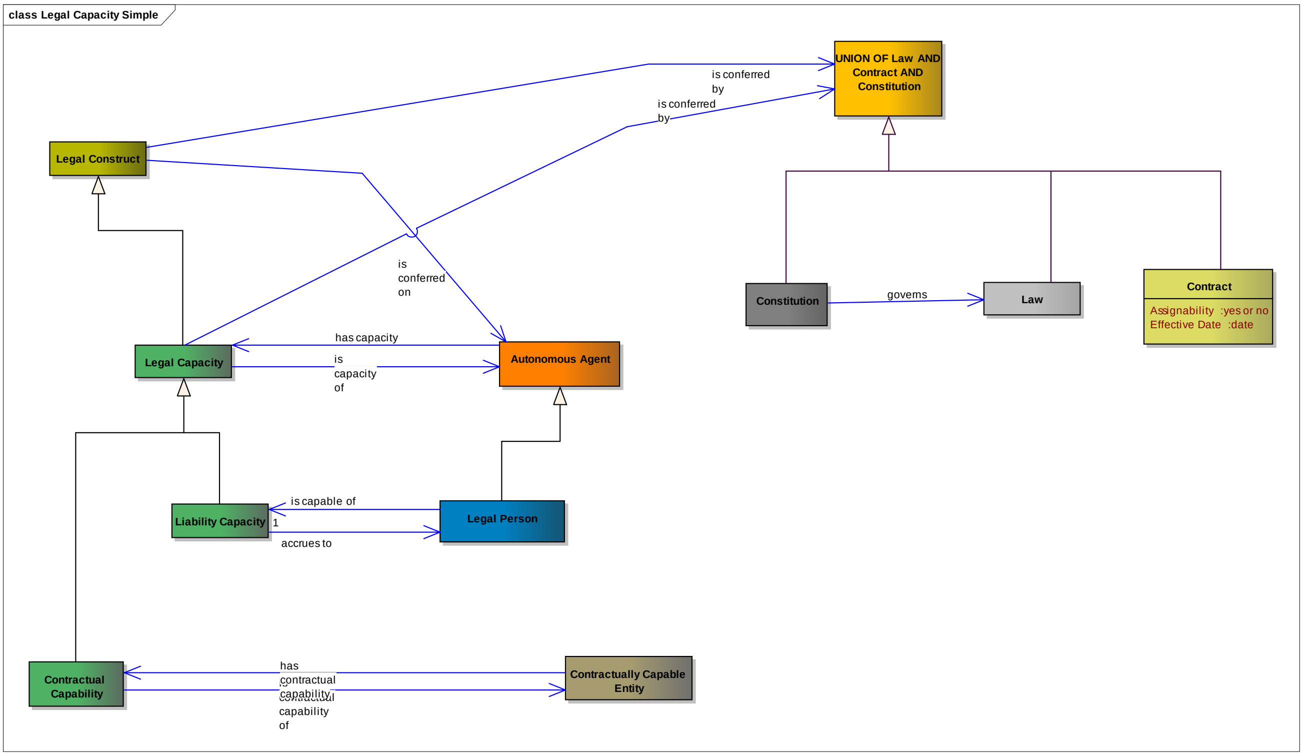
Legal Constructs

C.4.4.4 Statute

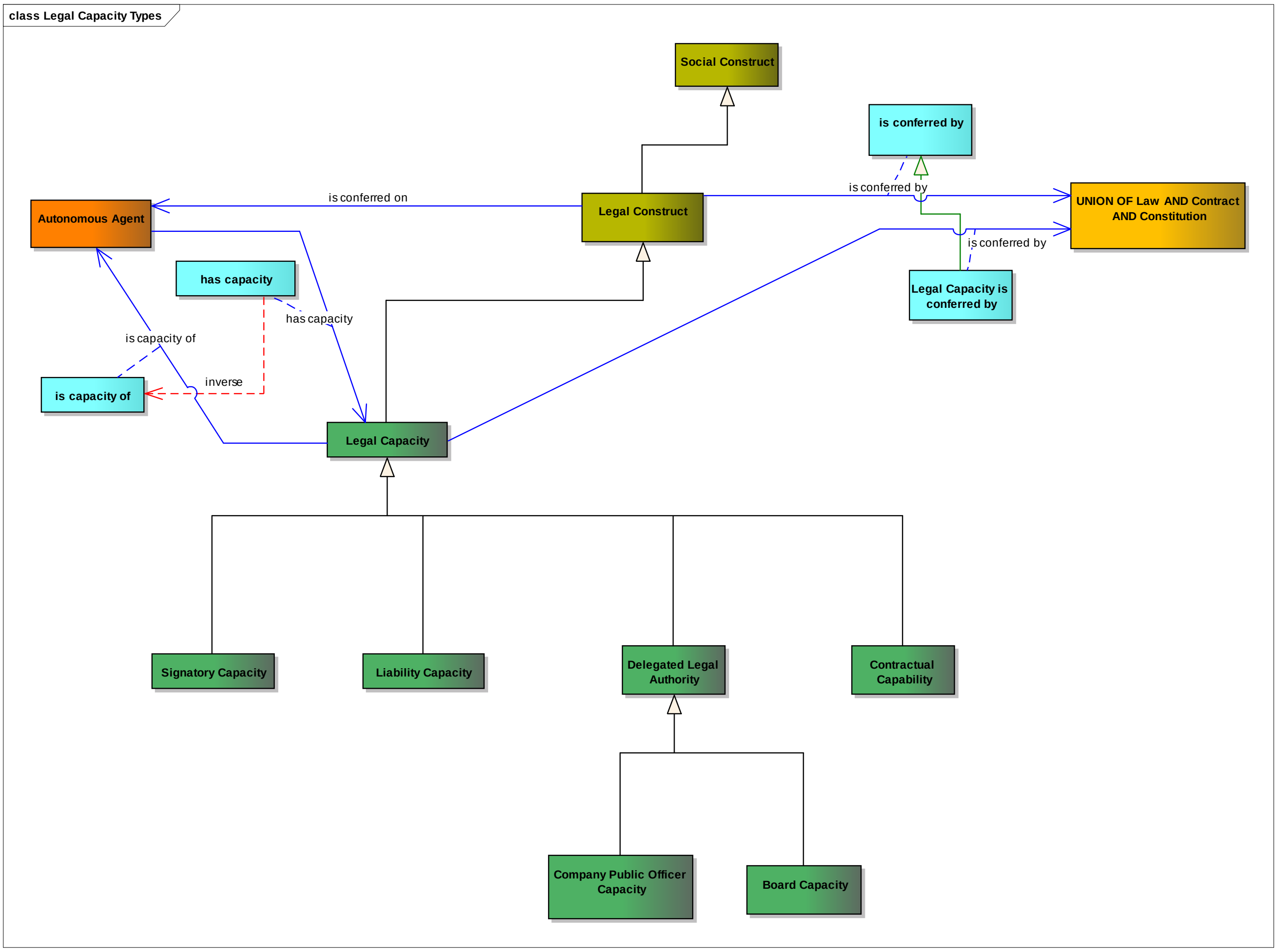


Statute Terms Overview

C.4.4.5 LegalCapacity

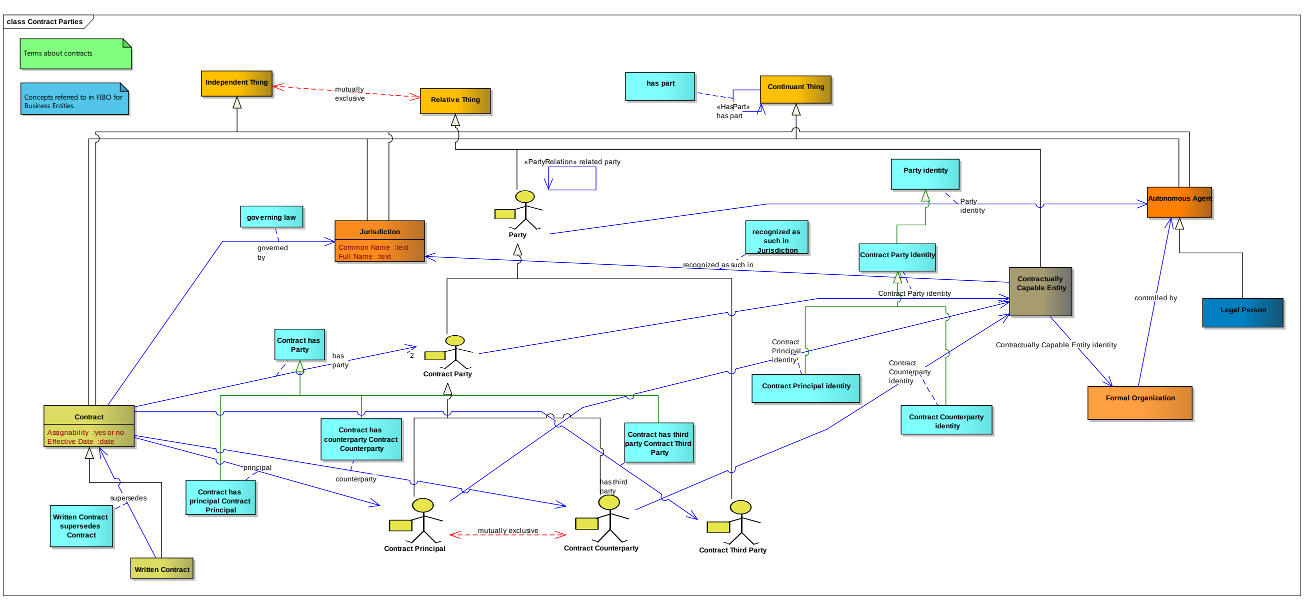


Legal Capacity Overview

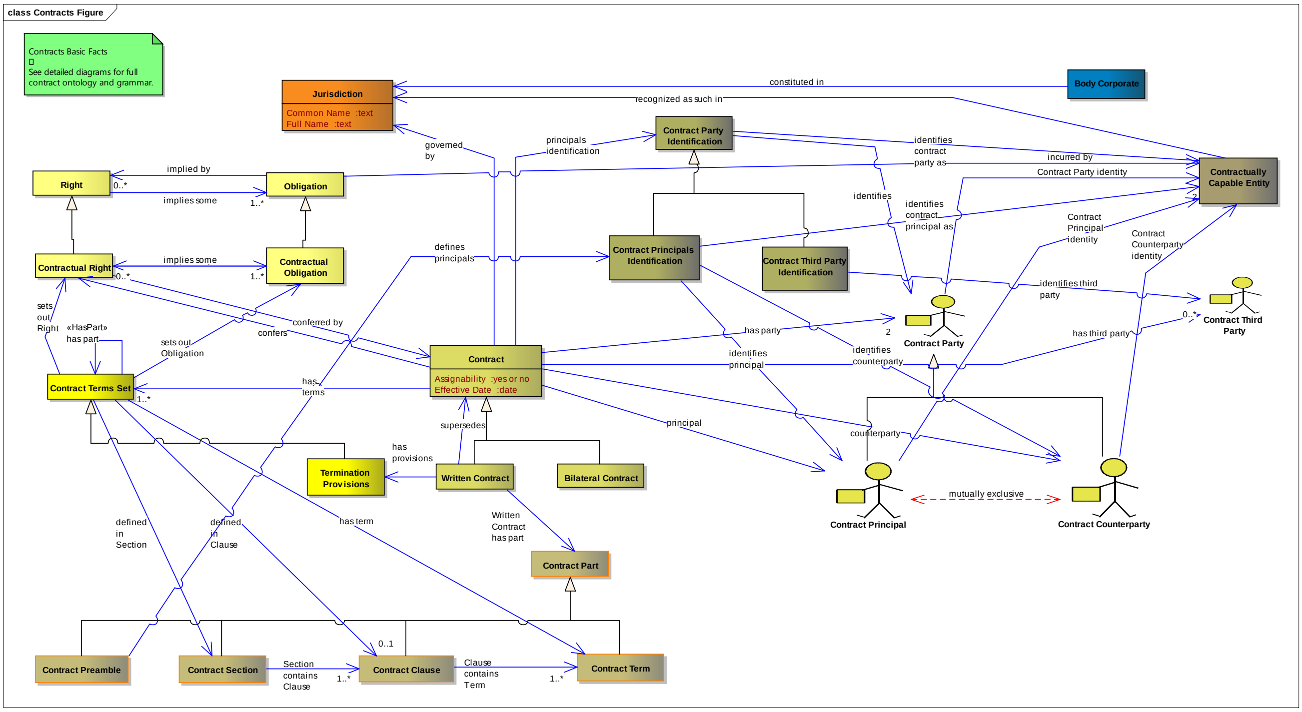


Legal Capacity Types

C.4.4.6 Contract



Contract Parties



Contracts Figure

C.4.5 Information

These are included because certain of the “Information” ontology terms are referred to in some of the annotation metadata (specifically the Provenance annotations, which refer to documents and standards from which terms and/or definitions may have been sourced in the development of FIBO content).

C.4.5.1 InfoCore

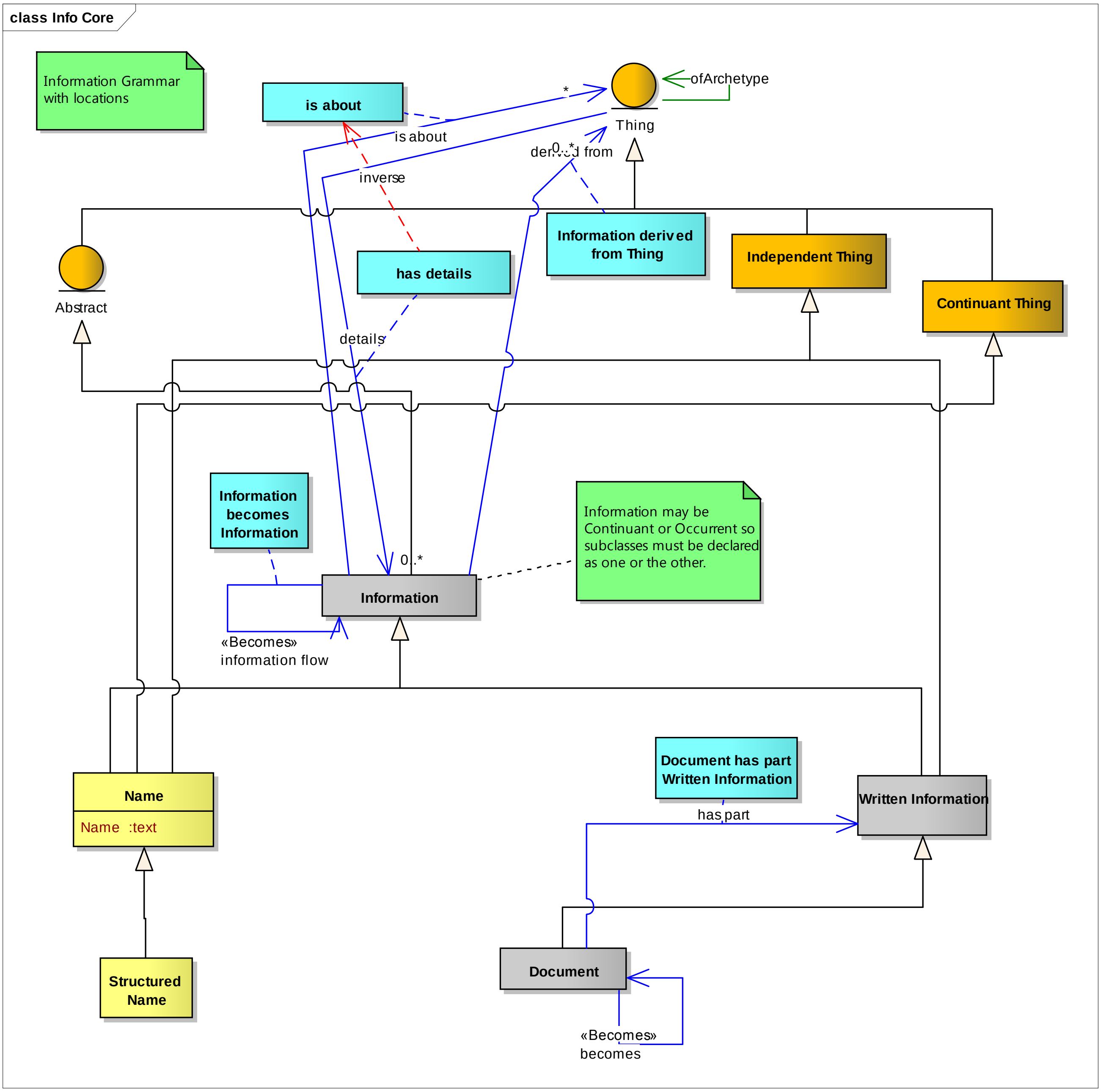
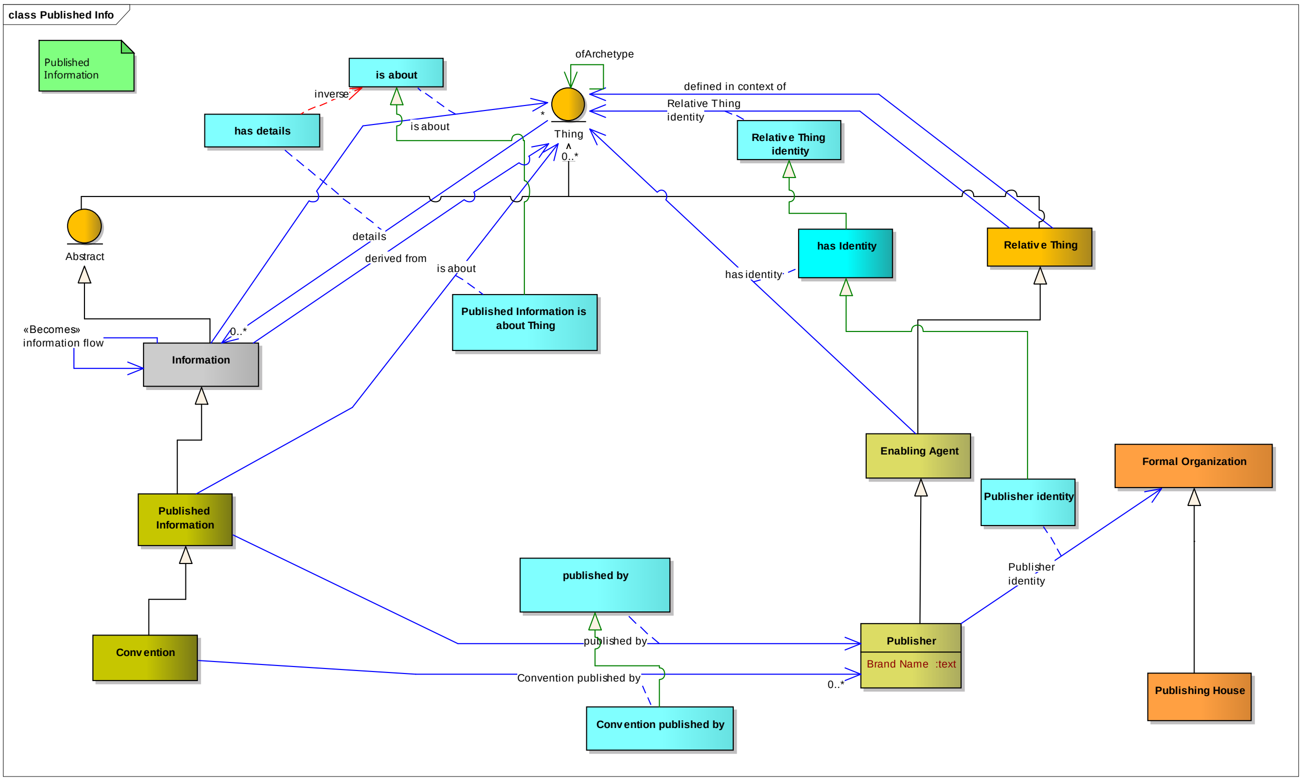


Diagram showing the main "Information" terms currently referenced in business ontologies.

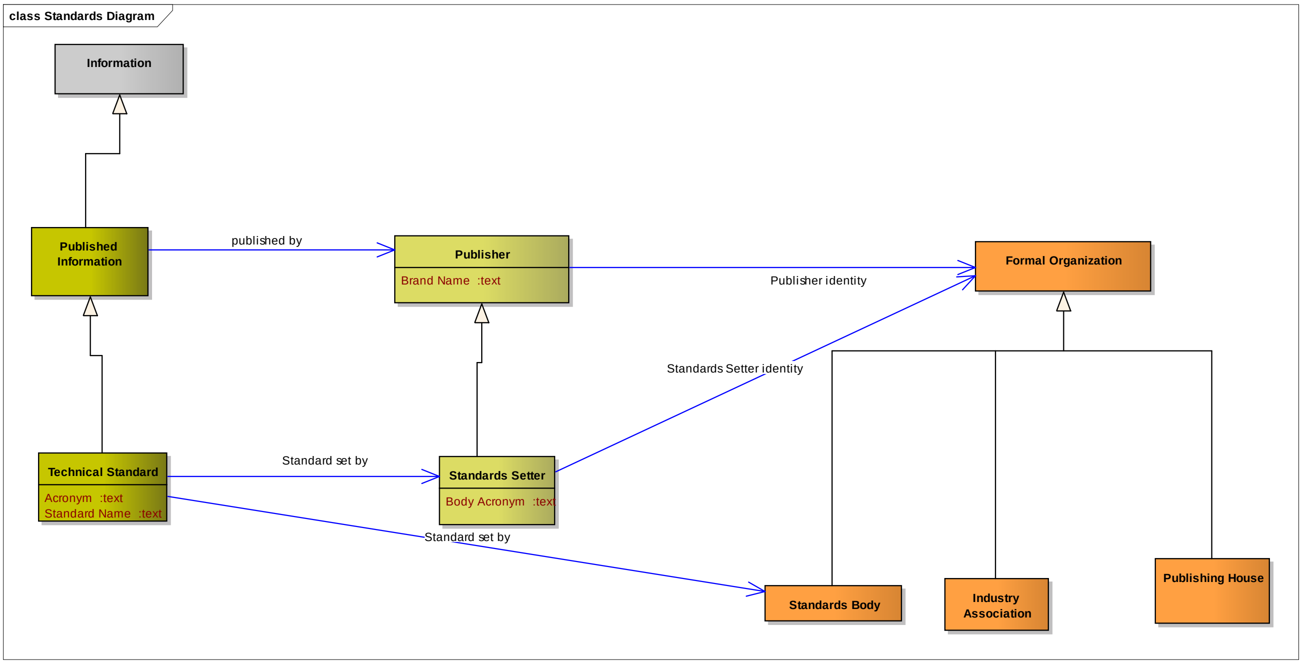
C.4.5.2 PublishedInfo



Information which is published by some business entity or publishing house.

C.4.5.3 Standards

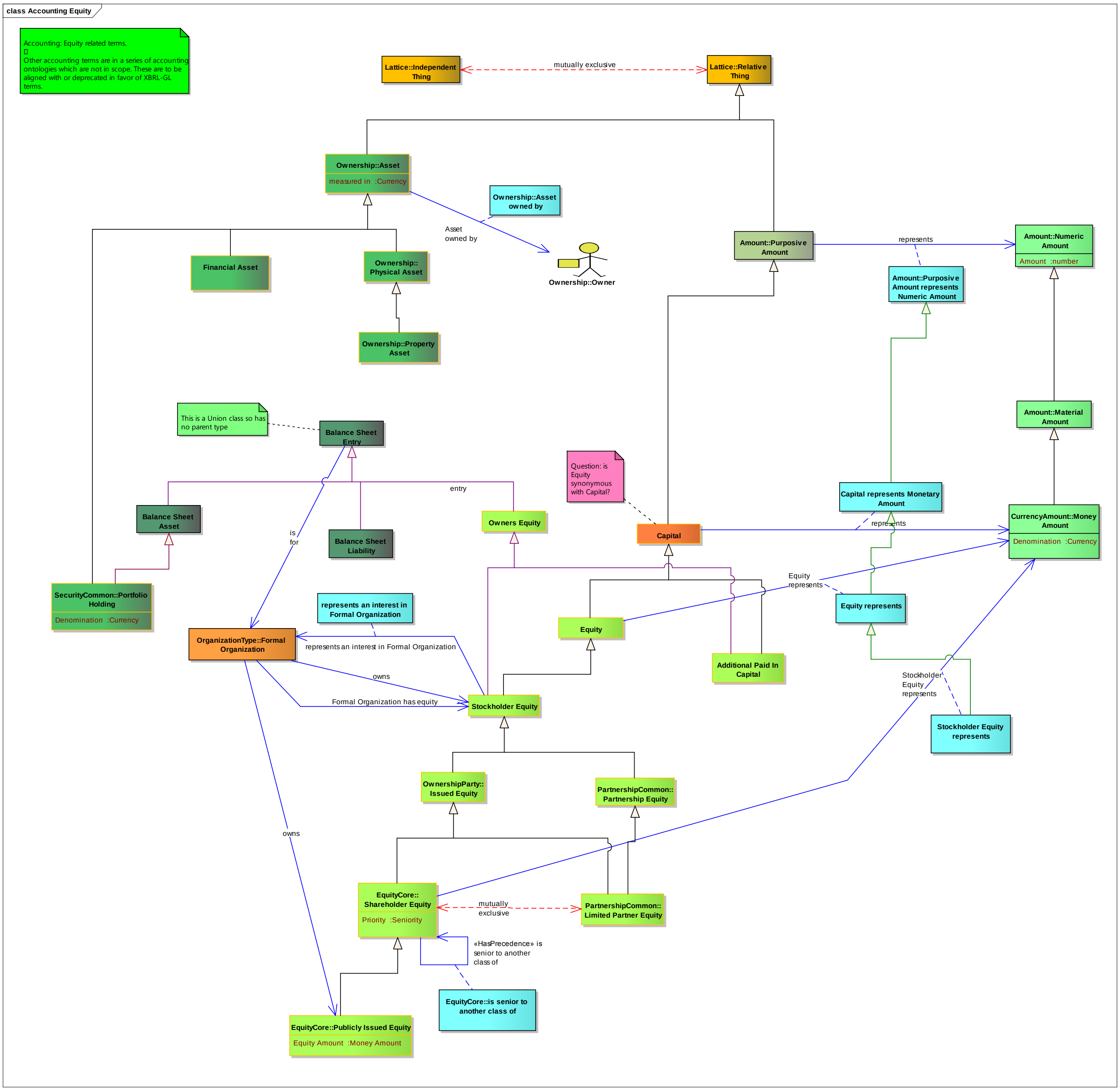
This ontology and the related StandardsIndividuals ontology exist so that it's possible to refer to individual standards bodies such as ISO, as well as to frame some other conventions such as derivatives exercise dates conventions and date roll conventions which may or may not have a publisher. The StandardsIndividual ontology is not included in the current scope as it is only needed for securities and similar terms.



Published information conventions (individuals not shown – see next diagram)

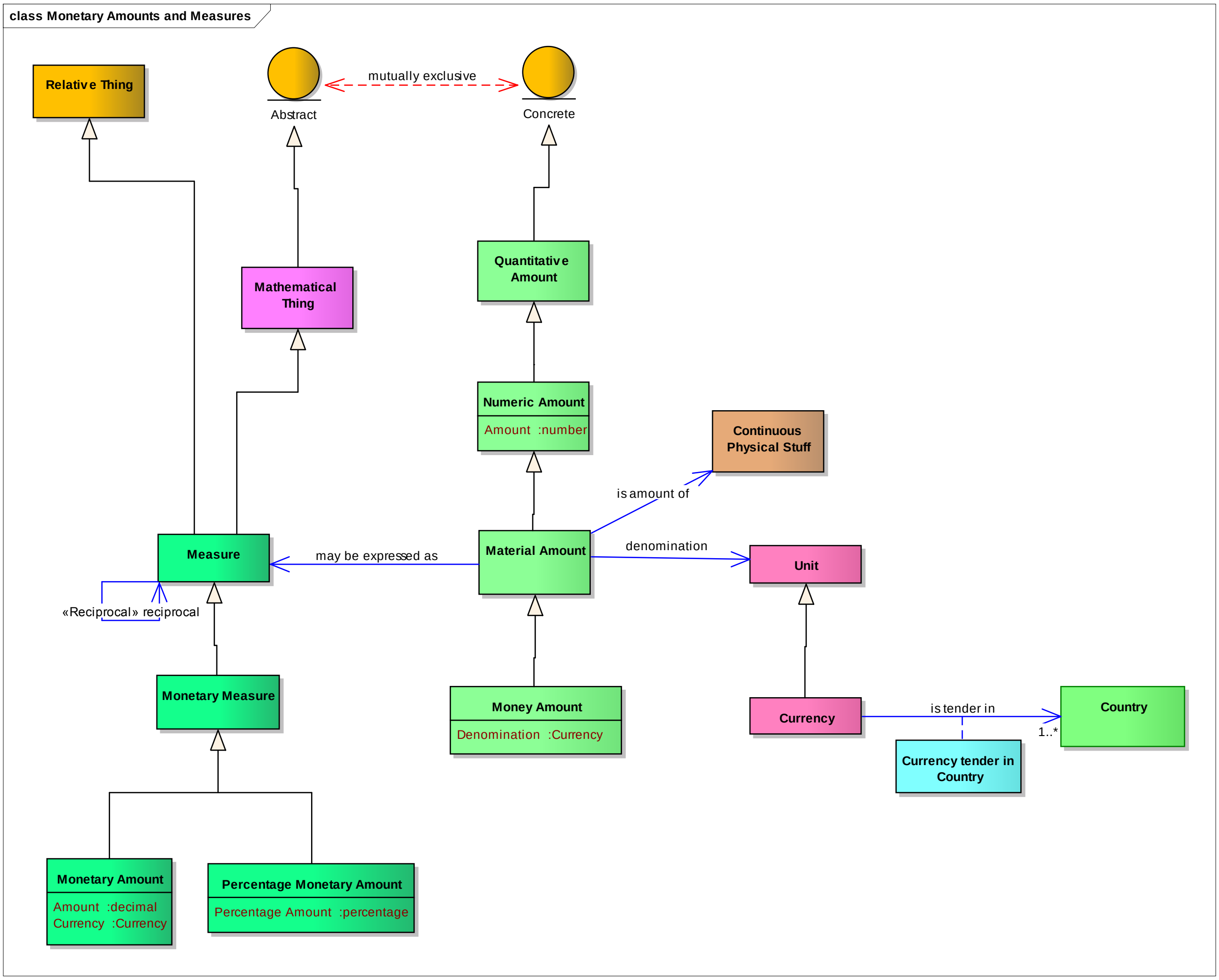
C.4.6 Accounting

C.4.6.1 Accounting Equity



Accounting: equity related terms

C.4.6.2 CurrencyAmount



Monetary amounts and measures

Annex D: Shared Semantics Treatments

(normative)

D.1 Introduction

Audience for this Annex: Semantic Modelers; Technical audiences.

The model content is grounded in terms which come from outside the realm of business entities of financial services. These are maintained in the sections titled 'Global Terms'. Wherever possible, terms in this section are cross referenced to terms set out by suitable standards bodies and academic bodies, so that the meanings of these terms are grounded in a broader community of semantics modeling.

Some of these external standards are in the form of formal ontologies, modeled typically but not necessarily in the Web Ontology Language (OWL) and in any case grounded in formal first order logic. In addition, some terms are derived from models which are not formally grounded in first order logic but which in some way or another are identified as meaningful concepts, either by explicit mark-up of the model content, by some separate theory of meaning, or by some statement at the level of the model identifying it as a semantic model. Such models are typically in the Unified Modeling Language (UML) or some other formalism such as that of the eXtensible Business Reporting Language (XBRL).

Note that formal reference to terms specified in the XBRL family of standards is outside the scope of this specification as there are no terms in the Business Entities content which make reference to these; these will be covered in a future specification.

Some of the models so referred to are only referred to in part, for example because the scope of the standard, as identified by its use case, is very different to the scope of the terms we wish to refer to in the Global Terms sections, or because the ontology contains formal axioms or facts which are at odds with the definitions of the terms as we have them.

This section describes the range of treatments by which such external standards are cross referenced in the Global Terms sections. A number of such treatments have been identified, depending on the nature of the standard or vocabulary referred to in our Global Terms, the language in which it is framed or the extent to which we are confident of making direct formal reference to it. For example, for some ontologies we wish to make direct, explicit reference, whereas for others we may have less visibility or confidence in the maintenance arrangements of that model's content and so have elected to create a local 'snapshot' of that ontology with its own namespace.

D.2 Shared Semantics Treatments

Case 1: Complete, stable OWL Ontologies

Treatment: Create a surrogate of the ontology using ODM.

Because this is in ODM, it shall have the actual URIs of the external standard. The material in this model represents a direct surrogate of that ontology.

Case 2: Ontology Snapshot

If the external ontology is in OWL but we want to make a snapshot if it at a point in time

Treatment:

- Create clone copy of the ontology in our repository
- Allocate a URI which identifies this as a clone (to include the elements of the original URI plus "/fiboclone/")

- Use OWL Equivalent Class, to point from something in our ontology, to something in that ontology.

Note that for many ontologies, an alternative arrangement used is that of the Named Graph (please see separate section on this). In the case of Named Graphs we do not need to use OWL Equivalent Class but incorporate the elements from the Named Graph directly.

When to use snapshot

This is used when for any reason we don't want to track changes.

Case 3: Partial Snapshot

This treatment is for when the external ontology has a broader or different use case and range of concepts, such that we may not wish to refer to or replicate them all.

Treatment: Create a clone of the parts of the ontology we wish to refer to.

Otherwise the treatment is the same as for Case 2.

Case 4: UML Models

This treatment is followed when the external material is in a UML model which is not explicitly modeled with some formal semantic extensions, but the model itself is presented as representing meaningful concepts and not logical model design constructs.

Treatment:

- Create a direct copy of the UML model,
- In the UML model, replace Associations and other relationships with UML AssociationClass throughout
- Create relationships which are instances of the “citation” construct specified in the Annotation Metadata section:
  - Classes: Use citation instance from the class in our model to the class in UML
  - Object Properties: Use citation instance from the object property in our model to the AssociationClass class icon in the target UML
  - Datatype Properties: Not applicable. Only derive classes and relationships from external standards

In the event that there are constructs in the UML model which do not represent meaningful concepts, these will not be referred to be any such citation relationships. The aim of the citation relationship is to identify where we have determined that the meaning of the concept in the FIBO Global Terms section, is defined by some competent authority.

Annex E: Logical versus Conceptual Models comparison

(informative)

Audience: Technology Management

E.1 Comparison Table

The principal differences between a logical data model and a semantic model are shown in Table E1.

Table E1

Logical Data Model	Semantic Model
Represents elements in a database design	Should not include design information but is a model of business concepts
Represents data model design components (Classes in OO design; tables in relational database design)	Represents "Things" using set theory concepts
Combines common data structures for reuse and efficiency	No efficiency considerations because it is not a design; reiterates concepts as they apply
Single inheritance hierarchy	Multiple inheritance
May define a number of optional properties of a class, such that the application developer would know whether these apply or not	Defines what facts are applicable to a given type of thing.
Uses enumerations to quality classes	Enumerates classes ("Things")
Closed World Assumption (CWA)	Open World Assumption (OWA)

These are explained further in the sections which follow.

E.2 Detailed Models Comparison

Design Elements versus Business Concepts

A logical data model represents the design of some data structure such as a database or a message design. This differs from a physical data model in that it is not specific to any one implementation or platform. That is, a logical data model is a kind of "Platform Independent Model" or PIM, as distinct from a "Platform Specific Model" or PSM.

While a logical data model is not specific to any one physical implementation, it does represent some design. That is, the logical data model, like any logical design, represents the results of some design effort by some designer.

A semantic model does not represent any design of any solution, but explicitly represents facts about the problem domain.

If a designer sets out to design something, there should normally be something that they are working from. In the design of software, designers work from formal business requirements statements, such as "Use Case" models or a requirements specification document. For data, the equivalent is a semantic model. That is to say, a designer of a data model should be expected to work from some source of knowledge of the items which are to be catered for in the database or messages for which they are carrying out the design.

Components that are Represented (Classes, Tables or Things)

In order to create a model which represents the logical design of some database or message scheme, the modeler will create a model which represents components of that design. For example, in a relational database they will create a model of database tables, along with relationships between those tables, public and private keys and so on. A logical representation of the design is therefore a representation of database constructs, namely tables, relationships, keys and so forth. The logical data model design is therefore couched in a notation which has formal representations of those elements. This may take the form of an Entity Relationship Model (ERM) or an object oriented model in the form of a Class Model in the UML design notation.

Depending on the model notation chosen by the developer therefore, the model may be an ERM model of data entities and relationships, or a UML class model of classes, associations, composition relationships and so on. These are the items to which elements of the model refer.

By contrast, a semantic model does not represent a logical design, and the things in the semantic model represent instead the real world entities in the business domain itself.

For example, a logical data model for securities may contain a representation of data tables for data about shares, bonds and so on, whereas a semantic model of the securities domain will contain representations of shares and bonds themselves, as kinds of "Thing".

The relationship between a semantic model element and the things it represents is made explicit in the Semantic Web "Web Ontology Language" or OWL notation. In an OWL model, every kind of "Thing" in the model (also known as "Classes") is a set theory construct which defines membership of the set in terms of the properties of its members. All classes in an OWL ontology model are sub-classes of a class known as the "Universal" set, commonly labeled as "Thing". This is the set of which everything is a member. In this way it is made explicit that everything in the model is some thing.

Reuse

It is sensible when carrying out data model design, to identify similar sets of terms and combine these into reusable sets. A semantic model may end up combining common concepts if the concept can be described as a more general, more abstract variant of the kind of thing. However, this is not a requirement for model design - things may be combined according to similarity in the data structures without reference to their meaning.

This is really another aspect of the basic fact that, since a semantic model is not a design, it has no design constraints (note this may not the case for an individual semantic technology application, where constraints are rightly applied but are very different to those for relational database or message design).

Single versus Multiple Inheritance

A limitation of some (though not all) relational design environments and notations is that the classes would be arranged in a hierarchy of classes. These would be in a single inheritance "tree" i.e. each class has only one parent class of which it is a specialization (ignoring polymorphism for now).

Semantic models more closely reflect the real world dispensation of taxonomies of kinds of thing, namely that a set of classes may defined according to more than one property. For example, a whale is both a marine animal and a mammal according to two different kinds of classification hierarchy, and an individual whale, being a member of the class of things which are a whale, is classified as both kinds of thing.

This is particularly valuable in modeling of kinds of security for different applications. For example risk management and securities trading performance analysis have different requirements, based on asset types, cashflow behaviors and so on. One application would need to classify things according to one set of requirements. Regulators have different requirements to traders, and even different regulators or different areas of regulatory analysis and systemic risk analysis may dictate different ways in which the universe of instruments may be "sliced" for analysis.

Optionality

In standards, particularly message standards, it is good practice to have a number of properties that may or may not apply to a given category of data element (for example, for a data element for a debt security), and make all of these optional. This is practical: for any debt instrument, not all the properties necessarily apply, but someone wanting to send a message from one point to another will be able to populate the message with those properties that exist for that security.

This, by definition, does not represent the knowledge that business practitioners may have about what facts necessarily must apply for a given instrument of a given type. In order to provide a message which is complete and correct, the sending party needs to apply knowledge from outside the model, about what facts necessarily apply to a given instrument. This intelligence would typically need to be built into the application that builds the message which is sent according to that schema. The knowledge is not represented in the schema.

At base this is simply another way of saying that the logical design of the message is not a representation of the knowledge about the instrument. Needless to say, this is not a criticism of such a message, it is simply a statement of why the message schema is not a record of the knowledge about the instruments.

Enumerations

A valid and good design approach to different kinds of thing is to provide a single data element which is an enumeration, containing entries for each of a number of entries that distinguish these things.

In a semantic model, each thing in the enumeration is a separate class of "Thing". The presence of enumerations in a model indicates that this is a logical model.

Note that for simplicity is it sometimes the practice to provide an enumeration (of textual strings, or 'literals') in a semantic model. However this is usually a pointer to the need to develop the semantics of the model further.

Open versus Closed World Assumption

- Open World Assumption: Absence of evidence is not evidence of absence
- Closed World Assumption: Absence of evidence is evidence of absence

A closed world model such as a database is built with the assumption that there is data available for each field defined in the database for a given record. An open world model does not make this assumption, and so facts may be asserted whether or not there is data to correspond to those facts. This is what gives a semantic model the capability to express facts which define things.

What this means in practice is that facts can be asserted about a thing in a semantic model without consideration to whether these facts are represented by actual data. For example, a fact about any event is that it has a cause, however causes of events need not be known or represented.

On a more detailed level, a semantic model can describe and represent facts about things without those facts being represented as data. Very often the facts which define the nature of a thing may not correspond directly to data. For example, many financial instrument types are defined in terms of the legal rights and obligations that they represent to one or other party to the contract. These rights and obligations may correspond indirectly to data elements, but the legal facts themselves may be more abstract, i.e. a fact stated in terms of "has right to" or "commits to" may refer to the abstract concept of a right, while the data may contain details of those rights and obligations, which may be regarded as a sort of signature revealing the existence of those rights and obligations.

This would be true of anything which is defined and classified according to facts which are themselves abstract. This would include most legal concepts.

E.3 Model Partitioning

The "Global Terms" section is partitioned into several non-mutually exclusive categories, in the sense in which the term “partition” is used in the semantic modeling community. These are:

- Independent, Relative and Mediating things
- Concrete and Abstract things
- Continuant and Occurrent things.

Each partition is represented as a class of OWL Thing and as a sub-type of the OWL Thing class, without additional archetype indications.

Terms defined in the model in this specification, and any terms defined in future additions to this specification or in local ontologies derived by extension of this specification, may not have a direct parent class of 'OWL Thing'. All classes of thing in the model described in this specification are given a parent which is either an archetype class of Thing or has an archetype as an ancestor, and all archetypes are given a parent from each of the three partitions listed above, with the exception of temporal terms which exist in a separate partition to the above.

Users of parts of this model may optionally ignore the above partitions in order to dispose model content under separate partitions of their own.

E.3.1 Independent, Relative and Mediating Things

This set of partitions provides a division into the model according to categories which have been arrived at through a considerable body of philosophical literature, notably that of C. S. Peirce. This partitioning relies on the claim in that literature that all things which can be named and classified fall into one and only one of these categories. This principle is reflected in the model described in this specification.

An independent thing is something which is defined in its own right and without reference to any context. For example, a business entity is an independent thing.

A relative thing is something the definition and meaning of which is specific to some specific context. That which is defined in that context is itself identified as some independent thing, or in some cases some other kind of relative thing, which stands in the role or relationship defined as the relative thing. For example a party to a contract is a relative thing, being itself some independent thing, in this case some business entity.

A mediating thing is the context in which some thing is defined as being some relative thing. For example, the context of contractual relationships, or of the context in which some specific kind of contract is entered into, is the mediating thing in which the business entity is identified as being some contract party. The term 'Mediating Thing' is synonymous with 'context' in the broadest sense of that term.

Relative things always have a relationship of 'identity' with some thing which may stand in the role identified by the relative thing. This is usually but not always some independent thing. In some cases the identity relationship may refer to some other relative thing, for example a securities issuer may be a 'Special Purpose Vehicle' which itself is defined as a kind of relative entity, the identity of which may be a company incorporated by the issue of shares, a limited liability partnership or some other form of legal entity. For this reason, while relative things should normally have an identity relationship to some independent thing, the most general application of this relationship is to the universal class 'Thing'.

E.3.2 Concrete and Abstract Things

This partition simply identifies whether something is a concrete item with weight and mass, or an abstract construct. Many of the concepts formally identified in the financial services industry are by their nature abstract.

Archetypes may only be identified as concrete or abstract if this is necessarily the case for all things of that archetype.

Note that things which have legal standing and which may be either provided on paper or in a dematerialized form are identified in this model as concrete. The intention of the Abstract partition is to define things which by their very nature are abstractions, such as goals.

One important class of abstract things is those things which are made up of information. According to the modeling principals, only things which are real may be represented in this model. This necessarily excludes things like database keys and locally defined identifiers. A common sense test needs to be applied to any kind of information before it is considered to be real and therefore able to be modeled here. Public information constructs such as security identifiers, business entity identifiers, credit ratings and the like pass this test because they are published by some party. In addition, documents and messages and the like which are passed between entities or parties in the course of carrying out some business process are equally real even though they are not published. The test for their reality is passed because information constructs such as documents have some real business, legal or financial import, that is some impact on something which is itself modeled as being part of the real world and not part of the technical design of some data or application.

E.3.3 Continuant and Occurrent Things

This partition segregates things which by their nature have some existence of a period of time, with a beginning and an end to their existence, and things which by their nature occur at a point in time. The precise timescales on which a thing may be said to occur or to have an ongoing existence is itself dependent on the domain being modeled, in this case all concepts relating to business entities and more broadly to the carrying out of business activities in the human world. So for example a human being would be considered on an astronomical scale as an occurrent thing, the difference in granularity in the time scales being determined according to the context in which the ontology is to be used. More precisely, a human being could still be considered as a Continuant Thing, with a human life being the corresponding Occurrent Thing, so in many cases it is reasonable to try to frame definitions of things which are clearly either continuant or occurrent.

For the avoidance of doubt, the partitioning of continuant from occurrent things is not formally represented by any axioms, and is definitional only. This means that terms in this model may be cross referenced to terms in models which use different formal ways of distinguishing continuant from occurrent things, for example what are called four dimensional, three dimensional, and similar modeling arrangements. The partitioning given in the model described in this specification contains no such assertions and is provided to enable the problem domain to be partitioned according to the basic nature of what is defined. This enables the model to contain concepts to do with events, processes, states and the like, though these are not utilized in the business entities semantic model.

Annex F: How to extend FIBO ontologies

(informative)

Audience: The intended audience for this Annex is semantic modelers, who are expected to have some familiarity with the basic principles of semantic modeling but not necessarily with the principles specific to FIBO. Basic OWL principles are also reiterated here. This section is not intended for purely business audiences or purely technical audiences.

This Annex should be read in conjunction with the section on Conformance (Section 2).

F.1 Terminology used in this Annex

There are several sets of terminology in use throughout this specification, and the meanings of some terms (such as 'thing') may be different in different specialized usages. Here the intended sense of these words, unless otherwise stated, is the sense used for business communication of the ontology content, and not the sense used in technical modeling or conventional Semantic Web terminology. If a formal definition of a term is not given or referred to via the "Definitions" section of this specification (Section 4), the normal, English language sense of a word should be assumed, and not that of any technical body of knowledge or community of practice.

The model described in this specification follows the principles of the Web Ontology Language (OWL). This defines the concept of a 'Class' as a set theory construct and is not to be confused with the usage of the word ‘Class’ in the UML modeling paradigm. In descriptions aimed as business audiences, we usually use the word ‘Thing’ in place of this, and on the basis that the OWL library class “Thing” is the ultimate parent of all classes in an OWL model (so they are all things). This also precludes having to explain to a business audience the very nuanced distinctions between UML and OWL Classes. The specialized technical usage of the word 'Thing' to refer to an OWL individual is not the sense used in this Annex.

In this Annex, the term 'class' and 'thing' will be used interchangeably to describe the OWL classes as set theory constructs, that is in the natural language (dictionary) sense in which one speaks of classes of thing (for example in the sentence "what class of locomotive is this?" or "what class of animal is a fish?"). This corresponds to the OWL usage of the term but not (or not without some qualification) to the UML usage of the term.

F.2 Overview

F.2.1 Classes of Thing

In OWL and therefore in FIBO models, membership of a class may be defined intensionally by way of properties which define the membership (the extension) of that class, or extensionally by way of listing the members of the set which makes up that class.

In the model described in this specification, all classes are defined intensionally except where extensional models are unavoidable. The modeling notation employed here supports the definition of extensional classes but this is discouraged except for the definition of classes which are necessarily extensional such as days of the week.

F.2.2 Model relationship to Subject Matter

The formal statement by which everything in the model has an ultimate super-class which is the universal set of 'Thing' is the means by which this model is formally identified as being a business conceptual model and not a data model representation.

In order to preserve the integrity of the model as a model of business concepts, all classes which are added to the model must:

1. Be given a superclass (a class with which the new class has a sub-class relationship) from one of the existing classes in the model;
2. Represent something in the business domain itself, and

3. Represent a set of possible members which in all cases would also be members of the set defined by the superclass in (1)

F.2.3 How to Model New Classes

In modeling semantics, it is a requirement to model each new kind of "Thing" (hereafter referred to as 'classes') in the model according to the following two criteria:

- What kind of thing is this?
- What facts distinguish it from other things?

The consequence of addressing these questions is that for each kind (or class) of thing in the domain of discourse (in this case business entities and legal entities), this will be defined in terms of the following question:

> "What is the simplest kind of thing that this is one of?"

By defining classes in terms of simpler kinds of thing, future changes will be additive. This benefit only applies if each class in the model is adequately generalized into some more abstract concept.

Failure to adequately generalize classes of "Thing" in the taxonomic hierarchy will have the result that future additions to that part of the taxonomy may prove to be disruptive. When the model is extended in the future to cover additional concepts, if the model components are not adequately abstracted then it will become necessary to break the existing chain of generalization to interpose new terms to support these new concepts. It is therefore important that modelers exercise imagination in this regard.

F.2.4 Declaring Class Disjointness

A disjointness relationship indicates that two classes of thing are mutually exclusive, that is that members of one may not also be members of the other.

Class disjointness refers to the situation whereby the members of one class may not also be members of another class when there is a disjoint relationship between the two. In OWL this relationship uses the 'isDisjoint' construct.

New 'isDisjoint' relationships should be labeled with the natural language label of "mutually exclusive"

Classes may have several separate sets of sub-classes which are mutually disjoint.

Note that disjointness is inherited through sub-class relationships. If a disjoint is misapplied this may cause inconsistencies. Conversely, if there is an inconsistency and disjointness has been correctly applied, then somewhere in the model there is an incorrect statement which would assert that some individual may be a member of more than one mutually disjoint class. The application of disjoint relationships therefore provides a useful diagnostic for subsequent extensions to the model, provided it is implemented correctly.

F.2.5 How to Model New Facts about Things

There are two kinds of "fact" in the model (in formal modeling terms, two kinds of "Property"):

1. Relationship Facts (known in OWL as Object Properties);
2. Simple Facts (known in OWL as Datatype Properties)

These are similar in their intent, in that they assert something about the class of which they are a property, but are shown differently in model diagrams.

Facts (properties) should be presented in the model only at the level of the class to which they apply. If a fact is not always applicable or relevant to the meaning of some concept, it should be applied to one or more sub-types of that class where it would be applicable. Similarly a property should not be applied to sub-classes where they would not always be true.

As an example, vertebrates are a class of things which are an animal and which have a backbone. It would not be appropriate to model the term "has backbone" as an optional property of all animals. Nor would it be sensible to say, for each class of things which is a vertebrate, that this class of vertebrates also has a backbone.

Note that there is a difference here from data modeling. In a data model it may be more efficient to assign a property to a class, make it optional, and then have some sub-classes which use that property and some which do not. This is appropriate for a data model because such a model is not intended to convey the meanings of those classes; rather, the user of the model has to know which sub-classes would have data for that property and which of them would not. In contrast, the semantic model in FIBO is intended to convey the knowledge that such a user would need to have. For this reason, considerations of efficiency which would be brought to bear on a data model design exercise, should not be considered when extending FIBO models.

Impact on Sub-classes

When adding a new relationship fact or simple fact to an existing class, ensure that this fact would be true of all the classes that are sub-classes of this class, and that are sub-classes of their classes and so on. If the meaning asserted by the addition of the new property is not necessarily true of all the descendent classes of thing, then it would not be correct to add it to this class. Instead it should be added to those of the sub-classes to which it does apply (that is, those to which it contributes something of the meaning of what it is to be a member of that class).

If there is a clearly identifiable group of those sub-classes for which the property is applicable, then it is possible that these could be grouped together as a new sub-class with that property. However, the addition of such a class, being as it would be interposed into an existing class hierarchy, should be handled with care - this constitutes a disruptive rather than an additive change, and will have different and more stringent change management requirements.

Adding a Relationship Fact

Wherever possible, a relationship fact should be a specialization of another relationship fact which is already in the model. When adding the relationship fact, the RDF construct "subPropertyOf" should be used to assert what is the parent property.

The new property should extend or refine the meaning of the parent property in some way.

It is also allowable to have more than one parent property. This is appropriate in cases where the meaning of one relationship fact is recognizably derivable from the meanings of two or more other relationship facts. This construction should be used sparingly and with care.

Types of Relationship Fact

In terms of the OWL language, there are a number of distinctions between kinds of relationship which may be asserted in this model. For example, it is possible to assert that a relationship is symmetric, or that it is 'functional'. Functional relationships are relationships where only one individual of the type that's shown as the range of the property, may be that thing.

In the UML modeling environment, the information about what kind of relationship a given relationship is, is provided by means of tagged values.

At present the terms distinguishing different types of relationship are not widely used in the model. If in doubt, relationships should be added without attempting to populate this information.

When adding a new relationship and making it a sub-property of some existing relationship, modelers should check the parent relationship and any of its parents, to verify whether these are defined as being one of these specialized types of OWL object property. If they are, then the new relationship will also take on this type, so modelers must ensure that this would be correct for the relationship being added.

Adding a Simple Fact

Simple Facts may only have a range (the object of the predicate) which is a simple information type or an enumerated data range.

The simple information types may be found in the model section "Business Types". These include concepts such as text, numbers, dates and yes/no answers.

Simple facts should not have ranges which are technical datatypes (the XML primitive datatype set or the datatypes made available within a UML modeling framework). XML primitive datatypes are allowable in RDF/XML based OWL ontologies, and would be used in an operational ontology derived from these models, but for the purposes of business understanding of the model these are all either given aliases (like 'yes/no' for boolean), or have more detailed types derived from them such as the various kinds of number.

There are no "Complex Types" in FIBO. For presentation purposes in different UML editing environments it is possible to consider rendering certain relationship facts (OWL object properties) as if they were simple types, i.e. using the UML "attribute" construct, but this is not formally supported in the sub-set of ODM defined in this specification. If this technique is used, such properties must be formally identified as OWL object properties; datatypes properties may not refer to classes which themselves have properties, such as monetary amounts or dated values.

F.2.6 Inverse Relationships

Whenever two relationships are in an inverse pair, this must be indicated by adding a relationship between those relationships, using the OWL construct 'inverseOf'. This should be labeled with the natural language label of 'inverse'.

Many relationship facts about things in the real world come in pairs, where one is the inverse of the other. For example "Account held by Account Holder" and "Account Holder holds Account" are two ways of saying the same thing, from the two perspectives of the Account and the Account Holder.

All relationships in the semantic notation used here and in the Semantic Web are unidirectional, that is they are 'triples' of the form Sub verb Object.

This is different to the way relationships are treated in data modeling. The 'ends' of a relationship in a data modeling format may be considered as being analogous to the separate relationships in a semantic model.

When to add these: Where it is considered relevant in defining the meanings of concepts, relationship facts (other than symmetric ones - see 'Types of Relationship Fact') may also be given an inverse. It is not a formal requirement to indicate all the inverses that may possibly exist. Such relationships should be present in the model and extensions to the model if the two senses are in common use, if they correspond to a named term for which there is a formal definition in use in the financial industry, or if relationship facts that are commonly defined for sub-types of the class that they are a fact about, are commonly specified or referred to in the opposite direction to the one which has already been specified.

For this reason, the addition of new classes of thing in the model, given that these specialize existing things, may sometimes require the addition of the inverse of some existing relationship fact, which was previously implied but not present as a property in the model.

F.2.7 How and When to Use Enumerations

There are two kinds of enumeration in the modeling notation:

- Enumerated Data Range
- Enumerated Class

Enumerated data ranges look a lot like enumerated datatypes in data models. However, these are used differently and will not usually correspond.

The 'Enumerated Data Range' construct should be used to enumerate possible data literals, that is pieces of text, numbers and so on, any one and only one of which may be the literal value of that datatype property for one instance of that class.

Where a data model enumerations may enumerate types of real thing and are frequently used to "flag" some class to say what kind of thing this is, this arrangement cannot be used in the FIBO semantic model. If a class of thing may be of several types, then these should be modeled as distinct classes, each of them a sub-class of the class of thing that they are all types of.

Where a class is to be defined by enumerating its members (extensional definition of the class), then the class itself should be modeled not as an OWL Class but as an OWL Enumeration Class.

F.2.8 Global Terms Usage

Because it was a requirement that classes of thing be abstracted to their simplest possible types, the modeling already carried out in FIBO necessarily required the creation of a set of classes which, by their nature, are not unique to business entities or financial services terms and definitions.

There is a second scenario in which terms are required which are not unique to financial services. This is when a relationships fact (OWL object property) about some business entity has a relationship to something which is not itself a concept unique to the context of the financial services sector.

The terms which are not unique to the financial services sector are maintained in a separate part of the model repository and are given a separate namespace. These are known as the "Global Terms" ontologies. Use of the appropriate terms in these ontologies is normative for this specification, but in many cases these ontologies are being evolved, improved upon and better aligned with other publicly available standard ontologies and with relevant academic work.

These ontologies are described in the notes on the "Global Terms" section. In Semantic Web terms, these are mid level ontologies. These are additionally supplemented by the inclusion of an "Upper Ontology" consisting of three sets of underspecified, high level partitions into which all model content is divided.

When adding new classes or relationship facts, modelers should seek out and select concepts from within the Global Terms ontologies which represent the terms they need to specialize or refer to. They should also recognize and adequately respect the 'Archetype' of that term, as described in Section 8.4.1. In particular, the ontology partitions under which the required archetype term resides should be inspected and understood, in order not to give rise to inconsistencies in the resultant ontology.

New general terms should not be added without first seeking the appropriate terms in these Global Terms ontologies or in some recognized external ontology, which must itself be cross referenced using one of the methods described in Annex C (Shared Semantics Treatments), in order to create the necessary relationships.

F.2.9 Content Creation Summary

In summary, there are two scenarios where classes of thing are needed in any ontology for business entities, for financial securities, loans, derivatives and so on:

- The kind of "Thing" which something is;
- Things which are referred to in facts about things.

The first question will lead the modeler to find a more general class of thing of which to make the new class a sub-class. This should be sought initially in the ontology which is being extended, and after exhausting this, in the appropriate 'Global Terms' section of the model, which must be inspected and fully understood before implementing the new sub-class ('is a') relationship.

The second question will lead the modeler to seek out the appropriate class of thing to which they need to refer. Often, but not necessarily, this will require the creation of some new class of thing. For example, a new class of 'Interest Payment Terms' might be appropriate in order to define a property of a new class of interest-bearing instrument which is defined by way of unique interest payment terms.

Modelers should look in the first instance for some class of thing which is exactly appropriate to the new relationship. For example, concepts like "Monetary Amount" or "Dated Monetary Amount" may be appropriate targets ("Ranges" in Semantic Web parlance) for more than one relationship fact about more than one class of thing.

In the absence of such a class, modelers should add a suitable sub-class of some existing class of thing which is broader in meaning but otherwise identical to the class to which the new relationship fact is to refer. In the interest payment terms example above, they would add a new sub-type of the class which is 'Interest Payment Terms Set' or perhaps 'Fixed Interest Payment Terms Set' or 'Bond Fixed Interest Payment Terms Set' as appropriate. This should be labeled with a suitably business-facing label which uniquely describes it within that ontology and which as far as possible reflects what is unique about its meaning (note that meanings do not follow from these labels, but that business comprehension of the model follows from their allocation).

Where a term is not available for specialization within the ontology which the modeler is extending, these are to be found in the 'Global Terms' ontologies, which have been created for the purpose of providing such terms. These are ontologies of things which are not specific to financial services. These include legal concepts like contracts, business concepts such as service provision, as well as an extensive set of concepts for times, dates, mathematical constructs, events and activities, and so on.

If a suitable general term cannot be found then it may be necessary to extend one of the Global Terms ontologies. This should be undertaken as a collaborative effort since this term will almost certainly be needed again in the future and by others. Such terms should be defined with formal reference to other, publicly available ontologies (these being defined either in Semantic Web formats or in some presentation, notation of theoretical grounding which makes it unambiguously clear that the terms in question are not part of a data model or other logical design).

F.3 Presentation Considerations

The presentation conformance requirements described in this specification are mainly a consideration for those creating or setting up editing environments in different modeling tools, and are not covered in this Annex. However, in the course of creating extensions to the model content there are a number of considerations which the modeler should keep in mind, as described in this section.

F.3.1 Labeling

All classes, relationship facts and simple facts should be given natural language labels. These should be rendered with spaces just as normal text is written.

These labels should conform to the following style requirements:

- Classes: Names should be in Upper Sentence Case
    - Abbreviations (if used) should be in their normal upper case rendition e.g. ABC.
    - Small words (of, and etc.) should also be capitalized (this is to enable technical users to compress the names without loss of sense)
- Relationship Facts: Names should take the form Subject predicate Object with the casing as shown
    - Subject and Object to have the full name of the classes themselves except where this is cumbersome
    - The predicate (verb part) of the relationship name should be in all lower case, with spaces
        - If possible, relationship lines (which are displayed in 'simple' diagrams that don't have the boxes that come with the relationship facts), should be labeled with only the predicate.
- Simple Facts: Names should be in Upper Sentence Case
- Other types of "Thing" construct (OWL Union Classes, Intersection Classes, Enumerated Classes and Enumerated Data Ranges) should follow the same naming convention as classes.

In addition to the above constructs, which define the terms in the business domain, there are a number of built in constructs which make additional statements, in set theory terms, about the classes and properties. These should be labeled as follows:

- Logical Union relationships: these are rendered using the UML construct of a generalization set (UML "GeneralizationSet"). Such sets have one name. This name should be a natural language label, with spaces and in lower case. The label should make clear the sense that it is a union relationship defining the logical union of the classes which participate in the generalization set, for example by ending the label with the word 'union'.
- Disjoints (OWL disjointWith): should always have the label "mutually exclusive"

- Inverses of relationships (OWL inverseOf): should always have the label "inverse"

F.3.2 Ontologies

These are implemented using the UML base class of 'Package'. Names for these should be in Upper Sentence Case. Wherever possible short or one word names should be considered.

F.3.3 UML Considerations

UML Diagrams

Diagrams are not transferred from any modeling environment into or out of the model repository. Diagrams are to be created by the modeler for presentation to business domain experts in the area in which they are working, or in the case of new submissions of the model content for future updates, to the wider community, and must be designed to be readable by business domain experts.

UML Notation

No explicitly UML notation should be present on any diagram.

The guiding principle here is one of language: any diagram which includes anything which belongs in or looks as though it belongs in some technical notation, will signal to the business reviewer that this diagram is in a language for which they have had no formal training. No matter how obvious the meaning of a diagram appears to be, the appearance of any technical notation means that it will appear to be something that requires some technical training to parse its meaning.

This means that

- no repurposed punctuation marks may be present on the diagrams. For example:
  - no curly braces and therefore no OCL
  - no guillemets - so stereotype indications must be disabled
  - no plus signs at the ends of relationships or next to attribute names
- UML class partitions that are unused (such as the operations partition) must be made invisible - either by manually resizing the class box until the extra line disappears, or by some other means;
- Exceptions may be made for relationship multiplicities, but the implications of these must be clearly explained to business domain experts who are expected to review the model content
- The Generalization arrowhead is an exception to the above: although this represents a technical notation (Generalization in UML), its meaning is more universal and can be explained to business domain experts ahead of any review. Such explanations must either reference Aristotelian syllogisms or be described in terms of the "is a" relationship with examples from natural taxonomy, depending on the knowledge of the business audience, but should not make reference to UML or words like Generalization or transitivity.
- Namespace indications: in some tools these are indicated with a double colon, which breaks the first rule above. Diagrams with these on may be created and maintained so that maintainers of the content can keep track of what is in what ontology, but these diagrams should not be considered as suitable for general business domain distribution.

Diagram Layout

Modelers should take care to lay out these in a clear and consistent way.

Generalization relationships should be laid out with the "arrowhead" pointing vertically upwards, in either the vertical tree style or direct style of routing. This is because this relationship, while technology neutral (it represents a basic Aristotelian syllogism), has to be explained to business domain experts and should therefore be presented in the same visual layout in which it has been explained, namely to represent taxonomic hierarchies with the most general terms at the top and the most specific at the bottom. These generalization relationships should never be drawn or found pointing downwards or sideways.

Where possible, the physical arrangement of the concepts in a diagram should try to follow the layout of the corresponding concepts in the archetype diagrams for those concepts.

Where large numbers of concepts are found in the same ontology, modelers should try to create separate diagrams which emphasize separate aspects of the subject matter (for example segregating contractual terms from legal obligations, or events from parties).

The relationship sub-property relationships are a particular hazard to creating clear, clean diagrams. However, these should rarely be shown to business domain experts. Where practicable, modelers are encouraged to create, for each separate thematic diagram, a set of three diagrams: one with all the material that needed to be modeled, one without the class component of the relationship facts, and one without the simple facts (compressing the class glyph as needed to remove the appearance of the attributes partition boundary).

Diagram Notes

Diagrams may also be decorated with informative notes. However, nothing of substance to the model content should be included in these, since these will not be retained when the model is transferred into the model repository or into other modeling environments.

UML Diagram Boundaries

As with notes, these may be included in business diagrams to aid in readability, but these UML boundaries do not form part of the model content and are not retained when the model content is transferred between environments.

UML Packages

UML Packages do not form part of the model, unless the package is stereotyped as an OWL Ontology.

OWL ontology packages may not be nested within other OWL ontology packages.

Modelers may arrange packages as appropriate for the usage to which they intend to put the model, and as part of this they may elect to make hierarchical structures of packages. Packages which are not stereotypes as OWL ontologies may be used for the purposes of such organization. Such packages may only contain other such packages or OWL ontology packages (that is, they should contain no loose classes or other constructs). Such packages do not form part of the model content, and will not be retained when the model content is transferred between environments.

No relationships between packages should be interpreted as, or created to imply, any relationship between ontologies.

All ontology imports must be explicitly modeled using the ODM "owlImports" construct. Each ontology should contain a diagram showing the full set of OWL imports required for that ontology, up to and including the "Lattice" ontology.

Annex G: Creating Applications with FIBO

G1. Introduction

G1.1 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.

G1.2 Operational Ontologies

To cover:

- When to replace an object property with a Boolean
- Shortening the inheritance hierarchy
- Using independent things without relative things
- Redefining Relative Things as Independent Things
    - This is valid when the context of the application matches the "Mediating Thing" that is the context in which the Relative Thing is defined
    - Example: Legal Entity is a relative thing but for an application whose scope is constrained to one jurisdiction or LEI issuer, it can be treated as an Independent Thing
- Use of property chains
    - May be a Conformant application for this – to be determined.
- Extraction of single-inheritance (monohierarchical) taxonomy
    - May also be conformant, as a sub-set of the FIBO material
- OWL Restrictions versus rdfsSubPropertyOf relations between multiple object properties.
    - Unless or until there is a suitable, FIBO-conformant representation of OWL restrictions to a business audience.

G1.3 Conventional Applications

To cover:

- Possible architectures
    - Fincore style – use of semantically under-specified classes, with enumerations to identify semantics

  - Other styles – see e.g. IBM POC “Semantic Data Model”, a direct rendition of the ontology with addition of database keys

- General
  - Booleans – don’t have mixed semantics in one Boolean (causes combinational explosions)
  - Text: when to collapse a chain of properties that end in a text field, with just an attribute that has text as a datatype
  - Combining pairs of object properties into one association – with the object property names as the labels of the ends of the association
- UML considerations
  - When to render object properties with a specific archetype, as UML Associations or Generalizations
  - Multiplicity
- Relative Things
  - These may be treated as independent classes when the context of the application matches the “Mediating Thing” that is the context in which the Relative Thing is defined
  - Example: Legal Entity is a relative thing but for an application whose scope is constrained to one jurisdiction or LEI issuer, it can be treated as an Independent Thing
- Localization within a part of the taxonomy
  - Patterns for taking a starting point within the hierarchy (e.g. MBS versus Bond versus Security), and navigating each of the object properties that apply at that level, navigating downwards (but not upwards) in the taxonomy of things that are the range of the object property, and defining these as the full possible scope of the model
- Extraction via Context
  - From a given “Mediating Thing”, navigate to each of the “Relative Things” defined in that context, and each of the “Independent Things” that may take on the “identity” property of those relative things – this should result in a set of all and only those things needed for the application
  -