JIRA 10 Proposed Resolution

**Issue Summary:** Property Domains and Ranges Actions

# Discussion

This issue resolution covers the following merged issues:

* FIBOFTF2-3 Property hasInForce is in wrong ontology
* FIBOFTF2-4 Property Characteristics
* FIBOFTF2-10 Property Domains and Ranges
* FIBOFTF2-64 Application of FinancialDates to date related properties

## Issue #10: Domain and Ranges Changes

This issue covers the provision of meaningful domains and ranges to properties which currently have no domain or range stated. In many cases this is done with reference to the informative and “philosophical” conceptual ontology which does not form part of the OMG submissions. In other cases, properties from Relations are moved to ontologies where their required domains and ranges are present. Some properties which were not in the Relations ontology nonetheless have a domain or range of Thing when something in their own ontology is more appropriate and so these are simply changed within those ontologies.

In some cases this work requires that the intended meanings of the properties themselves be narrowed from what was originally intended. The approach taken is that if it was not considered reasonable to include classes which were abstract enough to be the domains or ranges of these properties, then it should also not be considered appropriate to include properties which are so broad. Conversely, where there is clear merit in the meaning of a property being treated quite broadly, if the class cannot be added to the Foundations ontology then as a minimum there is to be an annotation indicating what the more abstract domain or range would be outside of the Foundations ontology.

In many case a property was thought of and defined with some very clear domain or range in mind (for example law or jurisdiction) but them moved up to Relations with a domain and range of “Thing:” just in case. The approach taken here is to reduce the domain and range to match what the intended use of the property was. If in future we come up with more broad properties these can be made parents of the ones we have here. In some cases it is the label itself which suggests a broader usage – therefore users of the model must pay particular attention both to the definitions and to the asserted semantics of very general looking properties such as ‘appliesIn’. As a matter of principle, no property in the FIBO Foundations ontologies is defined by its label, and these are to be regarded always as mnemonics to assist in remembering what the asserted meanings are.

### Reference and Referent

There is a specific issue with the newly introduced pair of classes Reference and Referent along with the property refersTo and its sub properties. While Reference is a valuable new addition, it would be incorrect to frame the range of refersTo and its child properties as “Referent” since the term referent is necessarily a context-specific or “relative” concept, while all usages of these properties are as between independent things. This is a result of not referring to the upper ontology partitions for independent, relative and mediating thing, which would have made this distinction clear. The range of the property ‘refersTo” is to be changed to Thing. Meanwhile the domains of the sub-properties of refersTo are to be changed to Reference, so that it is not left to the reasoner to conclude that this is the domain of these properties, and so that the meaning is clear to a reader of the model. There are also two properties which are inverses of two of these, and a corresponding change is needed in the domain and range of these.

There are some properties elsewhere in the ontologies, and some to be moved from Relations to those ontologies, which should also be framed as part of the Reference / refersTo ontology pattern. The changes are to be made in Relations ahead of refactoring those properties.

## Issue #3: hasInForce Changes

This issue resolution also covers issue FIBOFTF2-3 (“Relocate hasInForce to Jurisdictions”). This is one of a number of similar or related properties, including isInForceIn (this being an inverse of hasInForce), which relate to matters of jurisdiction and law at least as formally defined. While the definitions for these properties suggest an ambition to have them also refer to policies being ”in force” it makes more sense to constrain the meanings explicitly to the matter of having legal force - if this is deemed problematic it would be prudent to rename them “hasLegalForce”, “hasLegalFroceIn” and so on.

In the event these are to be moved not to Jurisductions but to LegalCore, which deals with the matter of law. The definitions are re-framed to make the meaning and usage clear as being a matter only of law (but including under the broad heading of law such things as by-laws, company by-laws and so on).

### The Properties constrains and isConstrainedBy

Alongside hasInForce and isInForceIn, similar moves are needed for constrains and isConstrainedBy. The definitions of these clearly link up law (in its most general sense) and autonomous agents. This pair are not used anywhere in Foundations (rather than a constitution being described as constraining laws, as originally modeled in the legacy ontologies; these are now defined using isGovernedBy, meaning that the constraints imposed upon laws by constitutions are covered in the semantics of isGovernedBy in these ontologies – so that the usage of constrains are limited to the constraining of individual actions by laws - if it is still used at all (it may be moot given the reapplication of isGovernedBy described above).

## Issue #4: Property Characteristics Changes

In the Beta1 and Beta2 specs there were a number of properties with characteristics, but these were not modeled in the VOM source models and there is no explicit record o them (they were not in the original legacy ontology). A typographical error in the syntax for their type means these were also not seen or tested by reasoners, and following the implementation of the resolution for Issue FIBOFTF2-18 these are now visible to reasoners. Some of them have restrictions on them in the model which prove to be incompatible with those property characteristics.

After review by the FTF it was agreed that only two instances of properties having their characteristics would be retained, these being the isTransitive characteristic for the properties isMemberOf and isPartOf.

A later review revisited this and let to the conclusion that a number of properties should have the characteristic ‘isFunctional’ set.

Note that isPartOf here corresponds to what it known in the literature as “proper parthood”, this being the transitive relation whereby any thing in the world may be broken down into parts which can in turn be broken down into parts. This is distinct from a separate treatment of parthood which deals with named parts or parthood as a role played by some thing (for example where the thing is a wheel, the role is plays may be the ‘part’ of the nearside front wheel of a car). That treatment is not currently present in these models and if it were added modelers would need to be clear about the distinctions between these two treatments of parthood since these are not incompatible treatment but merely distinct concepts. Therefore for property parthood the property characteristic is retained.

Because of limitations in the OWL language and the logical implications of the different constructs, some types of restriction usage are not allowed with transitive properties – for example a cardinality restriction cannot be used. There are some of these usages at present which need to be changed (these were previously masked by the syntax error in FIBOFTF2-18 which has now been corrected).

For example it would not be possible to say of a given organization that it shall have 2 or more members (a cardinality restriction) by using this property.

HOWEVER that restriction actually applies to the inverse, hasMember, which is not listed among the properties blessed or cursed with a transitivity characteristic

Arguably a member may have members, but if it does those are not necessarily members of the original – the member of a company is not also a member of a collective of companies, only those companies are, so therefore the isMemberOf property should NOT be transitive In fact, and for the same reason, nor should the hasMember – the individuals who form the members of some organization which participates in some larger organization, may or may not have the rights extended to them of that organization. Therefore membership is not to be regarded as transitive.

Independently we have agreed to remove the cardinality for hasMember to be a minimum of two autonomous agents, on the basis that these include organizations which while not being strictly organizations in the usual sense, are formal legal entities and may in some jurisdictions have only one participant (this is other than Sole Trader, which is simply a functionally defined business entity). So the cardinality is to be removed.

A full account of the required changes for the retention of property characteristics is as follows (From Elisa Kendall email 9 November 2014); note that following the decision to not implement transitivity for isMemberOf, some of these are moot and are recolored gray:

If we keep isMemberOf and isPartOf as transitive, the following changes

are required in order to ensure that the ontologies in FND remain

logically consistent: [grayed out where we don’t]

1. In the new Codes ontology, modify the restriction on the property

isMemberOf, which restricts the values of class CodeElement, from a

qualified cardinality restriction on exactly one CodingScheme to

allValuesFrom CodingScheme.

2. In the Organizations ontology, modify the restriction on the

property hasMember, with restricts the values of Group, from a qualified

cardinality restriction on a minimum of 2 AutonomousAgents to

someValuesFrom AutonomousAgent.

3. In the new IdentifiersAndIndices ontology, modify the restriction on

the property isMemberOf, which restricts the values of class Identifier,

from a qualified cardinality restriction on exactly one

IdentificationScheme to be a restriction on the property hasDefinition,

leaving the remainder of the restriction unchanged.

4. In the new IdentifiersAndIndices ontology, modify the restriction on

the property isMemberOf, which restricts the values of class Index, from

a qualified cardinality restriction on exactly one IndexingScheme to be

a restriction on the property hasDefinition, leaving the remainder of

the restriction unchanged.

**Conclusion:** No impact at all, all these changes were for the hasMember property or its inverse. However, we separately agreed that organization membership should not be restricted to 2 or more members, and so that hasMember relation (incorrectly described above as being for Group membership) will change as described.

## Issue #64: Financial Dates Addition

Datatype Properties previously framed in terms of dateTime and in Relations, are now to be replaced with object properties which are framed with reference to the Date (FinancialDates) class.

Therefore they also need to be in ontologies other than Relations, since FinancialDates imports Relations.

These are to be implemented as new object proeprties in the ontologies in which they first appear.

In carrying out this work, some inconsistency was found in the application of hasEffectiveDate, which has been used both for a relative thing (PartyInRole) and for Contract which I now a mediating thing (Agreement). The restrictions which are in the model would make party in role and contract appear to be the same kind of thing which they must not be.

To address this a new property is added for the temporal feature of PartyInRole. This will be similar to holdsDuring but will not be the same property (again this is already used for mediating things, namely Ownership and Control).

# Model Changes

## Model Changes Overview

These actions are carried out in the order given, in order to preserve dependencies. This involves changing between ontologies from time to time e.g. to delete properties only after other ontologies have been updated to use the moved property.

### The refersTo Properties in Relations

A recent change introduced new classes as domain and range of Reference and Referent for a small group of properties, thereby addressing this issue for those properties. Some additional changes are needed, for example to account for feedback on the use of “Referent”.

The overall approach to changes in Relations is as follows. These changes are done in phases:

1. Change the refersTo and its friends, to remove Referent, change domains of the sub properties to Reference

* Some of the properties to change under this issue may rightly belong under refersTo and not be in that set yet
  + Further analysis revealed a number of properties which are already related to the properties in that set but were not shown as such in the descriptions for the issue which introduced those or the accompanying diagram. Diagram updated to make these visible.
* Some of these could have gone to Arrangements – in the end they were kept here to minimize change

2. Do the AutonomousAgent changes (move 2 properties to Agents, reverse the import so that Relations imports Agents)

3. Create new versions of properties that are to be “moved” from Relations

3. Remove from Relations any properties that have been created elsewhere.

4. Complete the definition changes in Relations for the ‘entity’ relations definitions issue #13

## In AccountingEquity

* Change the domain of the property ‘representsAnInterestIn’ from Thing to Equity.

## In CurrencyAndAmount

Need to come back to this ontology after doing Percentage Changes 6 / 11 in Analytics – hasPercentageAmount range to be changed. Change the domain first time around.

Later: in fact this property is now removed. Note this will have considerable impact on IND, as indices expressed as percentages all need to use the new class-based percentages treatment. All of these should fail until they are refactored.

### Referring to diagram Physical Money Amount Concepts

#### hasAmount

* Add an annotation to ‘hasAmount’ as follows:

Annotation fibo-fnd-utl-av:explanatoryNote “The domain for this property should be read as being the term ‘Quantity’ which is in the informative conceptual ontologies.“

Annotation skos:editorialNote “If the term ‘Quantity’ is added to these ontologies in the future then that must be made the domain of this property – moving the property to Analytics or to Quantities as necessary at that time. “

#### hasCurrency

This is explicitly defined as referring to a monetary amount (not the base money unit of a money amount and not the denomination of some instrument etc.). Definition bears this out.

* Change the domain of the property hasCurrency from Thing to MonetaryAmount

#### hasNotionalAmount

Needs a note identifying what sort of thing may have a notional amount.

* Add an annotation to ‘hasNotionalAmount’ as follows:

Annotation fibo-fnd-utl-av:explanatoryNote “The domain for this property should be interpreted as being an abstraction which covers various forms of commitment, which may set out the existence of some notional amount of money, specified via this property. This is left unspecified for now, so that the property can also be defined directly as being a property of some contractual term which describes that commitment.“

### Referring to diagram CurrencyConcepts

#### hasBaseMoneyUnit

* Change the domain of the property hasBaseMoneyUnit from Thing to MoneyAmount

## In Jurisdictions

### Property appliesIn

Note there is no change to the definition, which already reflects the domain and range this is being changed to.

* Change the domain of appliesIn from Thing to LegalSystem
* Change the range of appliesIn from Thing to Jurisdiction

This also impacts the diagrams Civil Law Jurisdiction and Common Law Jurisdiction

### Property hasReach

The domain of this is too narrow – for some e.g. common law jurisdictions the rech may be geophysical as well as geopolitical (SME Review assertion).

* Add the class ‘Location’ (Locations) to the diagram
* Change the range of the property ‘hasReach’ to be ‘Location’ (Locations)
* On restriction fibo-fnd-law-jur-08:
  + Change the target of the someValuesFrom relation to ‘Locations’ (Locations)
* Change the skos:definition
  + From: “indicates the geopolitical entity (country, federal province or municipality) in which the jurisdiction has effect”
  + To: “indicates the geopolitical entity (country, federal province or municipality) or geophysical extent in which the jurisdiction has effect”

## In LegalCore

Properties to be moved from Relations:

hasInForce (change to range)

isInForceIn (change to domain)

isConstrainedBy (change to range)

constrains (inverse of isConstrainedBy and must be in the same ontology)

**Actions:**

* Add a property ‘hasInForce’
  + Add skos:definition: “relates a jurisdiction or situation to a rule, regulation or law (collectively ‘law’) that is currently in force in that situation or jurisdiction
  + Add element IRI
  + Metadata – there is no other metadata
* Add a property ‘isInForceIn’
  + Add skos:definition:”identifies a jurisdiction or similar context in which some law (including by-law, company by-law and state law) has effect”
  + Add element IRI
  + Metadata – there is no other metadata

## In Relations

Do the following changes in Relations ahead of moving some of the Relations properties into other ontologies, particularly Arrangements and Documents. This is to address issues with the Reference / Referent properties (see Discussion).

### In the diagram Reference and Referent Concepts and Relations

* Change the range of refersTo to Thing
* Change the domain of the following properties from thing to Reference:
  + appliesTo
  + characterizes
  + classifies
  + represents

The following is NOT changes to a domain of Thing, but to a domain of the class AutonomousAgent (introduced in a later step):

* + designates

See next step to reasoning and changes.

* For the following relations, which are inverses of two of the above, change the range of the property to reference, leaving the domain as Thing:
  + isClassifiedBy

The properties identifies and isIdentifiedBy are moved to Agents – so the above action is not carried out.

Retain the class “Referent” in case it is needed for models which explicitly call out the role of a thing as being the referent of some reference in the future.

#### Additional Actions (1) missing IRIs

Note that the classes Reference and Referent do not have an IRI following their introduction under Issue FIBOFTF2-20. Add the IRI for both of these.

#### Additional Actions (2) Missing Inverse Relations

Some of the relations included in the above change themselves have inverses which have not been included in the new pattern. These are:

* designates has an inverse of hasDesignation
* represents has an inverse of hasRepresentation
* denotes has an inverse of hasDenotation
* defines has an inverse of hasDefinition

For each of these: (except for designates for the reasons outlined below…)

* Change the range to Reference, leaving the domain as Thing

#### Additional Actions (3): misuse of ‘designates’ property

The property ‘designates, while it suggests that it refers to some reference which designates a thing, also has a sub property called “appoints”. This refers to an entirely incompatible semantic, the appointment (‘designation’ in that sense of the word) of some entity to carry out some role or have some responsibility. Either designates should not be in the new pattern introduced in FIBOFTF2-20, or appoints should not be a child of designates.

The definition for designates is: “to name something officially or appoint someone to a position officially”. Note this is a transitive verb (in the English sense) so it’s not clear whether the range should be the thing so appointed or the role to which the thing has been appointed. The domain is clearly the thing doing the naming.

The definition for hasDesignation (this being the inverse of designates) is: “relates an individual or entity to a position, role, or other designation”. Therefore if this is the inverse of the above then the party actively doing the designating of some other party is out of the picture and this pair of relations links some party which is designated to fulfil some role, and that role which it is designated (appointed, put in position) to fulfil.

The definition for ‘appoints’ is: “assigns a job or role to someone, selects or designates to fill an office or a position, fixes or sets by authority or by mutual agreement”

The property ‘appoints’ itself has an inverse: isAppointedBy. This has the definition “indicates the individual or group that has assigned or appointed someone to an office or position” – this is the third part of the set of actors in a designation or appointment context, and is not compatible with the usages given in designates v hasDesignation.

If we make designates and hasDesignation NOT inverses, then it’s possible for designates to be a parent of appoints, and not have any such relation between hasDesignation and isApopintedBy.

Then designtes and its child appoints links some autonomous agent doing the designation or appointing, to soe other automonous agent which is so designated or appointed 9but does not name the role to which it is appointed or designated). Then hasDesignation (not an inverse of designates) linkes the entity so designated, to the role or position to which it is designated (technically, thinginRole but modeled as Thing at this point). Meanwhile isAppointedBy is a simple inverse of appoints, and identifies who appointed some autonomus agent to some role but says nothing about the role to which they were appointed by that one.

**Therefore:**

Designates Domain: Autonomous Agent Range: Autonomous gent

Appoints Domain: Autonomous Agent Range: Autonomous Agent

hasDesignation Domain: Autonomous Agent Range: Thing (RelativeThing) or ThingInRole

isAppointedBy: Domain: Autonomous Agent Range: Autonomous Agent

The property hasDesignation is also a child of the property ‘has’. The sense of ‘has’ is intended to be ‘has intrinsic characteristic’, which would not cover the sense intended in appointment (but may cover the having of a unique representation called a designation).

**Conclusion:**

* No part of the intended meaning of designates belongs within the structure of “Reference” and “refersTo”
* The inverse of designates is hasDesignation but the inverse of appoints is isAppointedBy. These have very different senses and cannot both be inverses of properties in the same hierarchy.
* hasDesignation which refers to the role or other function of an individual (autonomous agent) would have a domain of AutonomousAgent, while designates, which refers to some agent designating another to some role, should have a domain of AutononousAgent. The range would presumably be a role of some sort, and should correspond to ThingInRole. We could keep this vague or invoke some reference to Relative Thing in the informative external ontology, however then no-no will know how to use it. Therefore it should be ThingInRole. However this is not available at this point in the model imports structure, so do a note instead.
* These cannot be inverses one of the other.

Therefore the following model actions are to be carried out :

* remove the parent relation between ‘designates’ and ‘refersTo’
* Introduce a new diagram, called “Designation and Appointment” and on this diagram:
* Move designates to this diagram
* Remove the subPropertyOf relationship between hasDesignation and has.
* Move appoints and isAppointedBy to the new diagram
* Remove the inverse relationship between designates and hasDesignation

### Other related properties

The following additional properties are found elsewhere in Relations and need to move to the Represents diagram:

* defines (a child of ‘represents’)
* hasDefinition is given as the inverse of defines and is a child of hasRepresentation
* hasRepresentation is a child of has
* hasDenotation is also a child of hasRepresentation
* denotes is an inverse of hasDenotation

These all appear as though they should be part of the Represents model area and are moved to the diagram introduced in Issue FIBOFTF2-20

Change the domain and / or range of these in line with the parent property “represents’ :

* change domain of ‘denotes’ to ‘Representation’
* change domain of ‘defines’ to ‘Representation’
* change range of ‘hasDenotation’ to “Reference’
* change range of ‘hasDefinition’ to “Reference’
* change range of ‘hasRepresentation’ to “Reference’
* Add the ‘has’ property to the diagram for completeness, showing which properties are sub properties of has.

Tidy up the diagram.

## Relations: Addition of AutonomousAgent to properties

At this point the class AutonomousAgent is to be made available by import of Agents into the Relations ontology so that certain properties can be defined as having a domain or a range of AutonomousAgent as appropriate and yet remain in Relations, causing minimal disruption to other ontologies.

Instead of moving AutonomousAgent into Relations, this is done by reversing the imports between Agents and Relations.

In addition there are two properties which have restrictions in Agents, so these need to be moved into Agents from Relations. Note that one of these, isidentifiedBy, is in the Reference / Referent set of properties and is the inverse of the property ‘identifies’. The intended use of this by restriction on AutonomousAgent is a little hazy – the original intention was not that an autonomous agent has some identifier but that it has some identity – a very different matter!

### workings:

Definition for isIdentifiedBy:

provides a unique identifier for an entity

Comment: This sounds wrong for Autonomous Agent anyway – though the restriction has minCardialiy of 0 to reflect that this isn’t always the case. It is right that this refers to Reference.

For identifies the definition is is the relationship between an entity and another that provides a unique reference for it

**Possible Solution:** Loosen identifies and isIdentifiedBy so that they do not refer to reference but to Thing, and let anything be the means by which something is identified (more in keeping with the sense for AutonomousAgent). Therefore remove the subClassOf relation from identifies to refersTo. It isn’t wrong but it deals with an otherwise circularity.

Then both of these should be added to Agents if they remain inverses of one another. If they don’t then isIdentifiedBy

Note also that a separate property, hasUniqueIdentifier, exists in relations and is intended to relate an identifier to the code itself as text. This is a datatype property and is not a child of isIdentifiedBy. This property is marked as functional to make this clear (it’s not the same as isIdentifiedBy). This is in Arrangements with usage in People for Identity Document and National ID number 0 in each case identifying the number associated with that document.

We will go with the “Possible Solution” identified above, and treat the identification of autonomous agents as an altogether broader affair than just them having ID documents and codes.

### Model changes:

#### In Agents

* Add hasName with domain of Thing and range of text
  + Add skos:definition: “that by which some thing is known; may apply to anything”
  + Add element IRI
  + Metadata… No additional metadata was given for the original term in Relations
* Add isIdentifiedBy with domain of Thing and range of Thing
  + Add skos:definition: “provides a unique identifier for an entity”
  + Add element IRI
  + Metadata… No additional metadata was given for the original term in Relations
* Add ‘identifies’ with domain of Thing and range of Thing.
  + Add skos:definition: “is the relationship be-tween an entity and another that provides a unique reference for it”
  + Add element IRI
  + Metadata… No additional metadata was given for the original term in Relations
* Remove the import relation from Relations (and all references in diagrams)

#### In Relations

* Import Agents

#### In Relations

Fix the designates relations:

* Change the domain of designates from Thing to AutonomousAgent
  + Note that since this is the domain, in VOM this has to be done by adding a proxy for AutonomousAgent into Relations

Close Relations, go to each of the ontologies which makes reference to identifies, isIdentifiedBy or hasName, before returning to Relations to do the next Relations step

#### Elsewhere: Refactoring for impact of Agents property moves

* Delete the properties that were introduced in Agents AFTER moving the references to this from all the other ontologies where they are referred to.
  + People
    - two restrictions on identifies
    - One usage of isIdentifiedBy
    - One usage of hasName (was not shown on diagram but is in model)
  + There are also changes to the diagrams in People from the changes already madein Reations for the refersTo relations that have a domain or range of Reference
  + IdentifiersAndIndices
    - One usage of identifies

### Back in Relations

* Delete pre-existing copies of the properties:
  + Identifies
  + isIdentifiedBy
  + along with their annotations and IRIs.
* Bring in the replacement hasName from Agents and movethe subClassOf relations to this from the existing hasName before deleting it. These are (in Relations):
  + hasFormalName
  + wasFormerlyKnownAs
  + hasCommonName
  + has Alias
* Then delete hasName and its metadata

**In diagram: Simple Physical Relations**

* Change domain of ‘holds’ to be AutonomousAgent (using local proxy in VOM)
* Change range of isHeldBy to be AutonomousAgent (Agents)
* Change domain of uses to be AutonomousAgent (using local proxy in VOM)
* Change range of isUsedBy to be AutonomousAgent (Agents)

**In diagram ‘Singular Relations”**

* Change range of isConferredOn to be AutonomousAgent (Agents)

Property isIssuedBy – intended to relate documents and other issued things, which are issued by some autonomous agent (including a country or government agency). Keep the domain general.

* Change the range of isIssuedBy to AutonomousAgent (Agents)

**In diagram “social Construct Relations**

* Change the range of isManagedBy to AutonomousAgent (Agents)
* Change the domain of ‘manages’ to AutonomousAgent (local proxy)
* Change the range of isProvidedBy to AutonomousAgent (Agents)
* Change the domain of ‘provides’ to AutonomousAgent (local proxy)
* Delete the property ‘hasInForce’ along with its annotations (1 definition 1 IRI)
* Delete the property ‘isInForceIn’ along with its annotations (1 definition 1 IRI)

NOTE: Most of the other properties on this diagram are to be shipped out to other ontologies.

**In diagram “The has Relations”**

* Make the property ‘isPartOf’ transitive (select isTransitive=true in tagged values)
  + Domain and range remain correctly s Thing, with no qualifying comments (this is proper parthood)
  + Add fibo-fnd-utl-av:explanatoryNote to this effect: “This property represents what is also known in the literature of 'proper parthood',that is the recursive (transitive) relationship whereby things have parts which have parts and so on. This is distinct from a separate meaning of 'has part' which would refer to an item playing the named role of a part such as a nearside front wheel. for the avoidance of doubt, this is not that relationship, and this property applies betwen independent things and other independent things which may make up their parts.”
* Add the following explanatory annotation to hasPart, in the same vein:
  + Add fibo-fnd-utl-av:explanatoryNote “This property relates a thing to anything which is a proper part of that thing. This is not parthood in the sense of a the role of part which may be played by interchangeable things such as wheels; instead this property relates an independent thing to something which makes up a part of it.”

## Explanations for More Abstract Domains and Ranges

* For the property ‘confers’:
  + Add fibo-fnd-utl-av:explanatoryNote: “This property should be read as describing the conferral of some legal power or duty, some commitment or some social construct, and is a property of some social construct such as an agreement or some legal authority. These concepts, which would describe the kind of thing of which this is a property, and the kinds of thing in terms of which this property is framed, are outside the scope of this mode land so are not shown.”
* For the property ‘isConferredBy’”
  + Add fibo-fnd-utl-av:explanatoryNote: “This property should be read as describing some legal power or duty, some commitment or some social construct being conferred as a result of some social construct such as an agreement or some legal authority. These concepts, which would describe the kind of thing of which this is a property, and the kinds of thing in terms of which this property is framed, are outside the scope of this model and so are not shown.”
  + Remove the skos:editorialNote which covers the same ground editorially.
* For the property ‘hasIdentity’
  + Add fibo-fnd-utl-av:explanatoryNote: “This property should be read as being a property of some kind of 'relative thing' as defined externality to this ontology. The property is usually but not exclusively framed with reference to some 'independent thing' but may take other forms and so should be regarded as having a target of 'thing'.”
  + [NOTE: This should not be a child of ‘has’ since this is not an intrinsic property of a thing but a relation between a relative thing and the thing which performs the role or function defined for that relative thing. This was not changed since there needs to be an issue to cover it]
* For the property ‘isMandatedBy’
  + Add fibo-fnd-utl-av:explanatoryNote: “This prooerty should be read as being a property of some social construct as defined in the informative ontology for conceptual abstractions, to some other social construct such as a legal instrument or an agreement.”
* For the property ‘hasMember’
  + Add fibo-fnd-utl-av:explanatoryNote: “This property should be read as being the property of a logical union of group and organization (not shown).”
* For the property ‘isMemberOf’
  + Add fibo-fnd-utl-av:explanatoryNote: “This property should be read as being framed in terms of a logical union of group and organization (not shown).”
* For the property ‘governs’
  + Add fibo-fnd-utl-av:explanatoryNote: “This property should be read as being the property of a logical union of social construct (in the informative abstractions ontology) and legal person, and as referring to 'thing'.”
* For the property ‘isGovernedBy’
  + Add fibo-fnd-utl-av:explanatoryNote: “This property should be read as being the property of some thing and as referring to a logical union of social construct (in the informative abstractions ontology) and legal person.”
  + Remove the skos:editorialNote which covers the same ground editorially.
* For the property ‘embodies’
  + Add fibo-fnd-utl-av:explanatoryNote: “This property should be interpreted as being the property of a union of concrete things and information constructs, and as referring to some abstract thing or to some mediating thing or both, in the informative ontology of conceptual abstractions.”
* For the property ‘hasContext’
  + Add fibo-fnd-utl-av:explanatoryNote: “This property should be read as referring to some context (known as 'mediating thing') in the informative upper ontology which is not included in this model. It should also be read as being the property of some contextually defined thing (known in the informative upper ontology as 'relative thing').”

### In LegalCore

* Add a property ‘constrains’ with a domain of Law and a range of AutonomousAgent
  + With skos:definition: “forces, compels, or obliges”
  + Add element IRI
  + Metadata: definitionOrigin: http://dictionary.reference.com/browse/constrains
* Add a property ‘isConstrainedBy’ with a domain of AutonomousAgent and range of Law
  + With skos:definition: “identifies the policy, rule, regulation, contract, or other thing that compels or obliges someone to act in some way”
  + Add element IRI
  + Metadata: there is no other metadata.
* Add an inverseOf relation from isConstrainedBy to constrains

**Discussion:** Constrains and isConstrainedBy are a pair, and currently in Relations. These are used to indicate when some entity’s actions are constrained by some “Law” in the most general sense e.g. statutes, by-laws, company by-laws. This is possibly used in Business Entities. It is used/not used in Foundations. In the legacy model this was used to describe how the formation of law is constrained by any constitution which is in place, but that usage has been dropped in favor of including such constraint under the semantics of ‘isGovernedBy’ in the current model. So it is possible that constrains, isConstrainedBy may not be used at all.

**Since**

* the stated domain of isConstrainedBy is explicitly an autonomous agent; and
* the thing which does the constraining is something under the broad (and existing) heading of law), so:
* the pair of properties should go in the LegalCore ontology. Note that AutonomousAgent is already available by import into LegalCore because the concept of CourtOfLaw required that of FormalOrganization which is itself an Autonomous Agent).

**Impact:** In Jurisdictions, change the reference from the restriction on Statute Law to refer to the new property ‘hasInForce’

### In Jurisdictions

In Jurisdictions, in diagram Statue Law:

* Change the range of the onProperty relation of restriction fibo-fnd-law-jur-06 so that it refers to the new property ‘isInForceIn; (LegalCore) and remove the reference to the old version of this property in Relations.

## Temporal Changes

### Effective Date and Expiration Date

#### Discussion

Original proposal that hasEffectiveDate and hasExpirationDate is to be replaced by holdsDuring:

This has been implemented for Ownership and Control (mediating things) and in principle should be applied to more of those

This is true for contracts – to be done – and toe Mediating Thing (context) in general.

However,

This is not implemented at present in People, where identity documents have an issue date and an expiration date. Unlike contracts, this is usually explicitly, and there’s no clear reason or cover) to remove this.

Therefore hasEffectiveDate and hasExpirationDate are not redundant as assumed when we looked at which properties are to be functional (these ones are functional in the OWL).

#### Model Changes

Dates are applicable to documents, to assets and to Arrangements.

Document is NOT given as a child of Arrangement therefore ates to do with Documents must be done in that ontology.

In the ontology “Documents”

* Import the ontology “FinancialDates”
* Add object property hasDateOfIssuance with domain of Document and range of Date (FinancialDates)
  + Add skos:definition: “links something, typically an agreement, contract, or document, with the date it was issued”
  + Add element IRI
  + Additional metadata (from original property): There is no other metadata
* Add object property hasExpirationDate with domain of Document and range of Date (FinancialDates)
  + Add skos:definition: “links something, typically an agreement, contract, document, or perishable item, with an expiration date”
  + Add element IRI
  + Additional metadata (from original property): There is no other metadata

Note that these definitions need to be revised – contract and agreement should be covered using the ‘holdsDuring’ pattern and these properties now apply only to documents (they needed to be somewhere other than Relations in order to apply Date from FinancialDates – this is part of Issue Resolution for FIBOFTF2-[Not yet raised], to make use of FinancialDates.

**Impact:** Table 10-28 “People Details” restriction fibo-fnd-aap-ppl-10 onDataRange dateTime changes to onClass Date

**Impact:** Table 10-28 “People Details” restriction fibo-fnd-aap-ppl-12 onDataRange dateTime changes to onClass Date

**In People:**

* For Property Restriction fibo-fnd-aap-ppl-10
  + replace the onDataRange relationship with an onClass relationship which points to the class ‘Date’ (FinancialDates)
  + move the destination of the onProperty relationship to point to the new object property hasExpiratonDate and delete the former hasExpiratonDate datatype property from the diagram.
* For Property Restriction fibo-fnd-aap-ppl-12
  + replace the onDataRange relationship with an onClass relationship which points to the class ‘Date’ (FinancialDates)
  + move the destination of the onProperty relationship to point to the new object property hasDateOfIssuance and delete the former hasDateOfIssuance datatype property from the diagram.

### hasAcquisitionDate and hasDispositionDate

These are superseded by the use of holdsDuring, as far as the acquisition, holding and disposition of an asset (which is what the definition of hasDispositionDate explicitly calls out). Therefore these are not needed.

These are left in Relations but unused – NO ACTION.

Note that hasAcquisitionDate is still seen in Ownership although hasDispositionDate is not.

THEREFORE

If this is used at all, then both should be used, specifically against Asset (definition to be reviewed later), and be reframed as object properties, in Ownership, with Date (FinancialDates) as domain. The alternative is to delete Restriction 01 on Asset and the use of hasAcquisitionDate (Relations) in Ownership ( in diagram Asset Definition)

*Meanwhile: why does holdsDuring (BusinessDates) have a range of Thing and not of DatePeriod?*

## Changes in Ownership

### isOwnedBy

formerly isAssetOf, this links the Asset (relative thing) to Owner (relative thing which is a partyInrole).

* Change the domain of isOwnedBy to be Asset
* Change the range of isOwnedBy to be Owner

## Changes for hasEffectiveDate in Relations

This is applied both to the PartyInRole class (in Roles) and in the Contracts class (in Contracts).

* It cannot be both.
* Agreements should now use the new holdsDuring relation as applicable for mediating things (as used for Ownership and for Control), and the hasEffectiveDate would then be moot
* The physical contract document has an effective date in the form of “hasDateofIssuance” which would also apply to the issue date of negotiable securities (these need a specific date not a treatment via the holdsDuring thing, since the data appears in all reference data!)

This appears to be an error in the specification as written – now that Contract is a child of Agreement, the use of hasEffectiveDate in both of these places would cause the relative thing PArtyInRole to be reclassified as the same kind of thing as Contract, a mediating thing.

### Proposal:

* Replace hasEffectiveDate in Parties with holdsDuring
* Create new hasEffectiveDate in Contracts, with range of Date (Financialates)
* Create separate new property in Parties.

### In Contracts

* Add a new property hasEffectiveDate with domain of Thing and range of Date (FinancialDates)
  + Add skos:definition: “the date a contract, relationship, or policy comes into force”
  + Add element IRI
  + Additional metadata (from original property) :
* On property restriction fibo-fnd-agr-ctr-03
  + Change the target of the onProperty relation from the old datatype property hasEffectiveDate to the new object property hasEffectiveDate
  + Delete the relationships “onDataRange”
  + Add a relationship “onClass” which points to Date (FinancialDates)

### In Parties

We need to replace hasEffectiveDate with something else appropriate to the temporality of PartyInRole.

If we use holdsDuring, this will (at least in principle) collide with the use of holdsDuring in Ownership and in Control, where it defines the life of the mediating thing (Ownership, Control).

However, the domain has been left as Thing (remembering to add an annotation for the interpretation of this). The problem is that there is no mediating thing in Parties, and the restriction is on PartyInRole

Model Change:

* Add a new object property to replace the usage of hasEffectiveDate here
  + Domain is PartyInRole
  + Range is Date
  + Name is hasCommencementDate
  + Label is ‘has commencement date’
  + Add skos:definition: “the date a party relationship comes into force”
  + Add element IRI
  + Other metadata: none (there is none in the original hasEffectiveDate)
* For property restriction fibo-fnd-pty-pty-02
  + Chane the target of the onProperty relation to hascommencementDate
  + Delete the onDataRange relation
  + Add an onClass relation which points to the class Date (FinancialDates)
* Delete hasEffectiveDate from the diagram

### Explanations for More Abstract Domains and Ranges in Parties

* For the property ‘isAPartyTo’:
* Add fibo-fnd-utl-av:explanatoryNote: “This property should be read referring to some context (known as ‘mediating thing’) in the informative upper ontology.”

## Changes in LegalCapacity

Intention is to reframe the property hasCapacity so it no longer has domain and range of Thing. Capacities are enjoyed by autonomous agents so this is the requirement.

* Add a local proxy for AutomonousAgent (Agents)
  + Add element IRI (this is the IRI of the original element in Agents)
  + Add the about relationship referring to AutonomousAgent (Agents0
* Change the domain of hasCapacity to be AutonomousAgent (using the proxy n VOM)

Change to definition of hasCapacity for Issue #13:

* Change skos:definition of ‘hasCapacity’
  + From: “identifies an entity that has some capability to carry out certain actions, or has certain rights or obligations”
  + To: “identifies an individual or organization that has some capability to carry out certain actions, or has certain rights or obligations”
* Change skos:definition of ‘isCapacityOf’
  + From: “identifies an entity on which a given legal capacity has been conferred
  + To: “identifies an individual or organization on which a given legal capacity has been conferred”

## Removals in Relations

Remove the following properties in Relations along with their annotations and element IRIs:

**In diagram: Social Constructs Relations**

* isConstrainedBy
* constrains

**In diagram: Data Properties Numeric:**

* hasEffectiveDate
* hasDateOfIssuance
* hasExpirationDate

## Property Characteristics Changes for JIRA Issue #4

### Make the following property transitive:

#### In Relations

* isPartOf

### Make the following properties functional (tag = isFunctional)

#### In People

* fibo-fnd-aap-ppl:hasDateOfBirth
* fibo-fnd-aap-ppl:hasGender

#### In Roles

* fibo-fnd-pty-rl:isPlayedBy

#### Originally In Relations

* fibo-fnd-rel-rel:hasAcquisitionDate
* fibo-fnd-rel-rel:hasDateOfIssuance – now in Documents
* fibo-fnd-rel-rel:hasDispositionDate
* fibo-fnd-rel-rel:hasUniqueIdentifier

## isMemberof cardinality

Although it was later agreed that isMemberof should not be transitive, we still need to go ahead and relax the cardinality in order to allow for the concept of an organization with one member.

### Model Checks:

In Parties, diagram OrganiztionMember: there is already no cardinality on any of these restrictions.

#### Elsewhere:

Organization (membership assertions)

Group itself: appears in FormalOgranization

* this is already not a cardinality restriction – no change

### In Organizations

* Remove the cardinality from restriction fibo-fnd-org-org-01
* Replace the onClass with an allValuesFrom

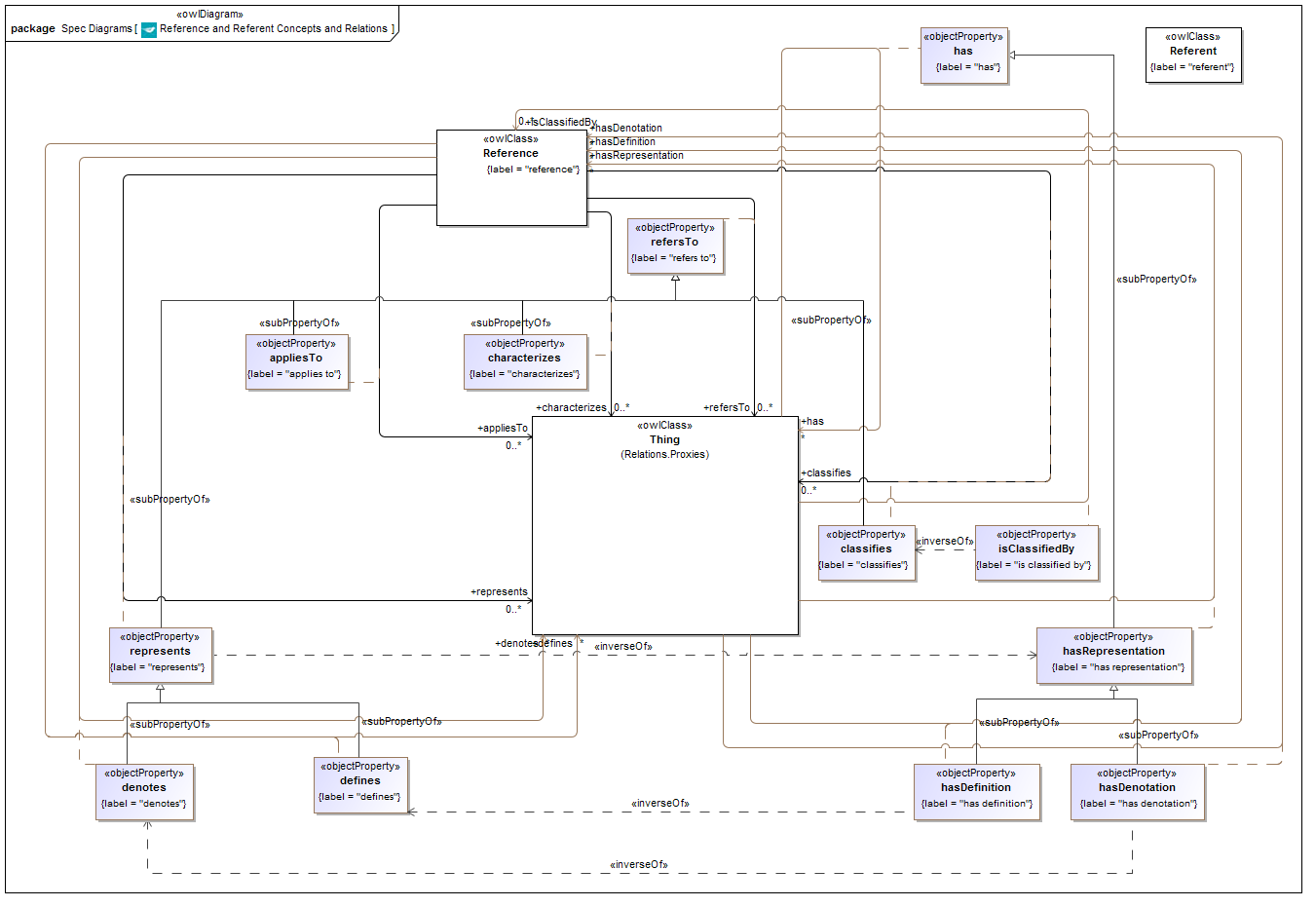
# Revised Diagrams

*[a later iteration of this document will put the ‘Chapter and Verse’ material here and add the tables; since this an autogenerated model those details will not ad to an understanding of the nature of these changes]*

## Module: Relations

### Ontology: Relations

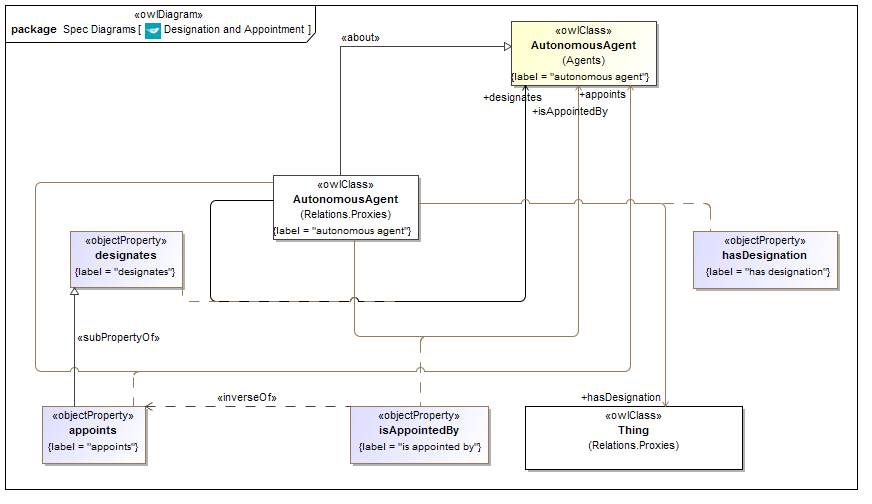
#### New version of diagram “Reference and Referent Concepts and Relations”



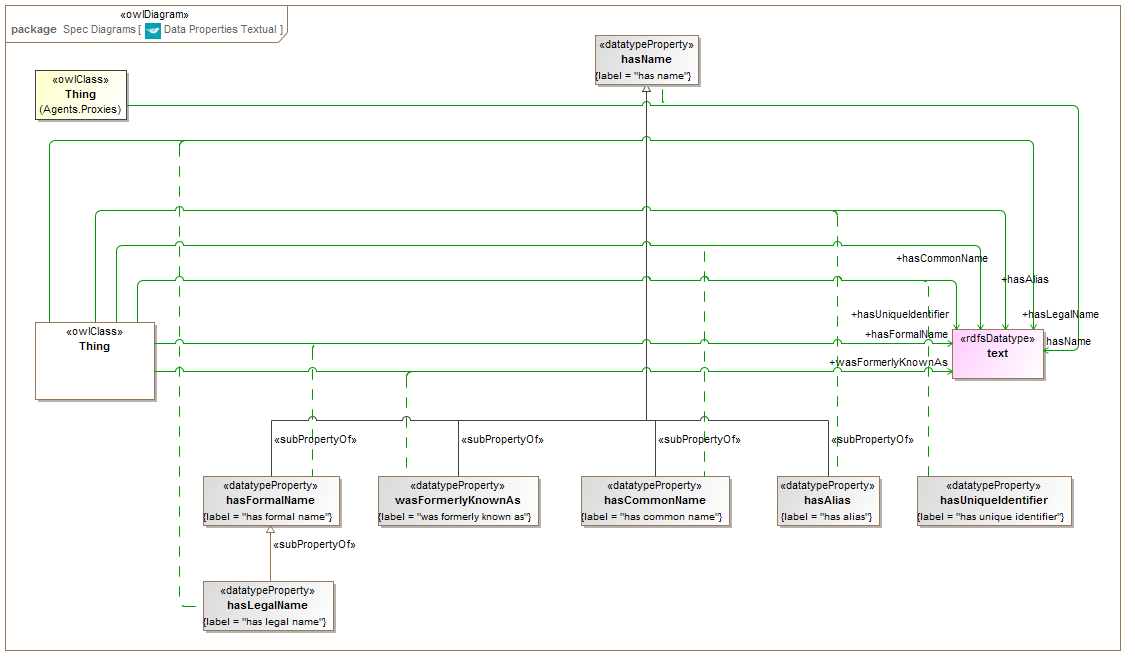
(shown after removal ot 2 properties that were moved to Agents)

#### Diagram “Information Relations” (Figure 10.12) is deleted.

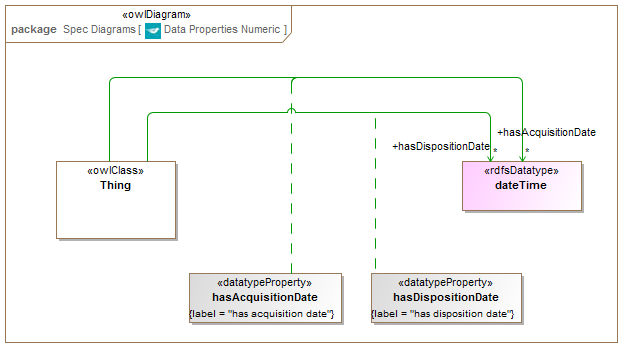
#### Designation Properties (new diagram)



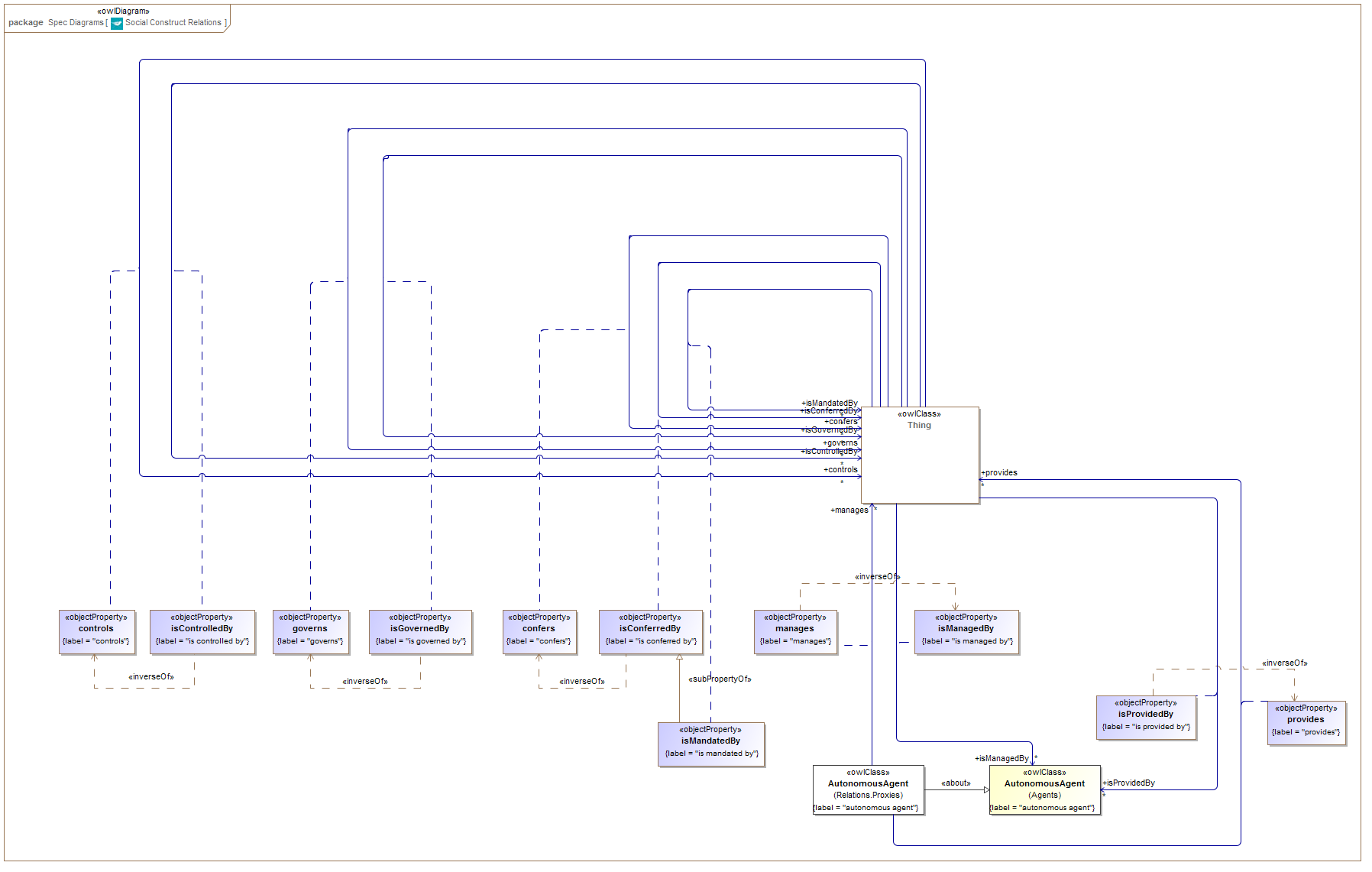
#### Data Properties (Textual)



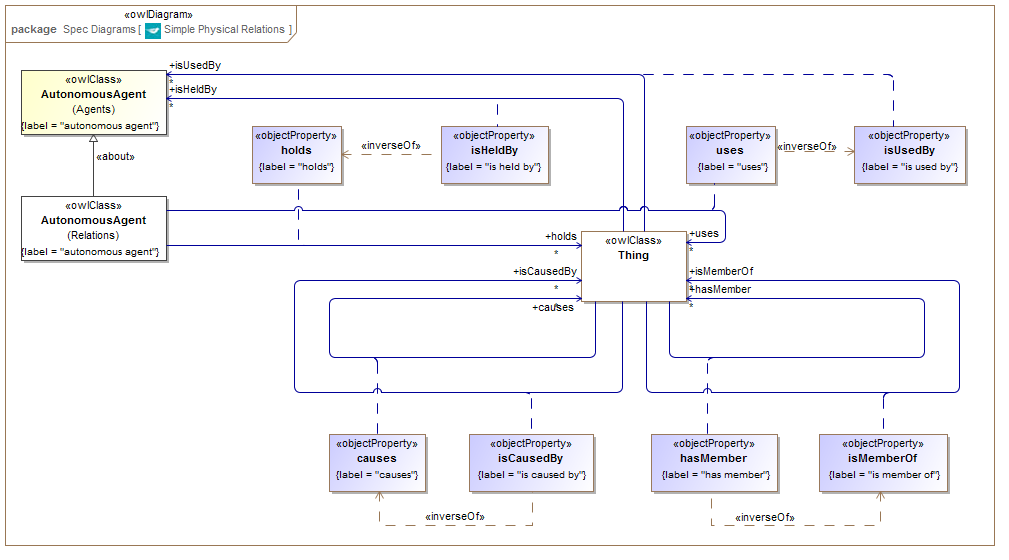
#### Data Properties Numeric



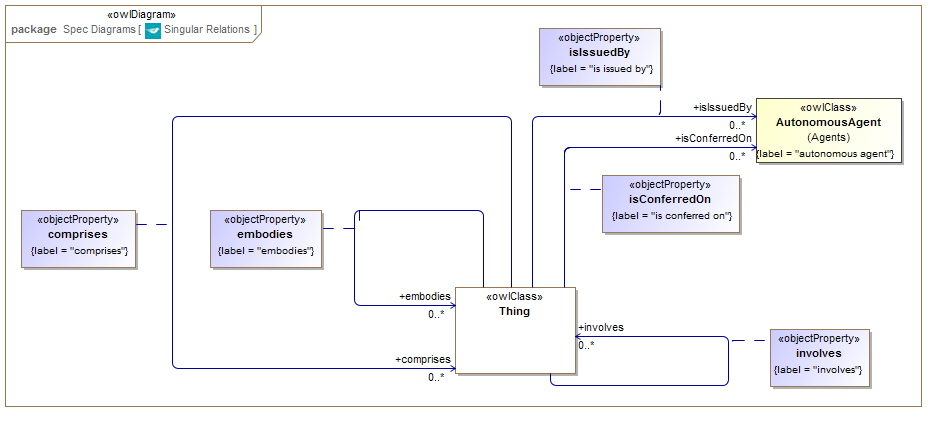
#### Social Construct Relations



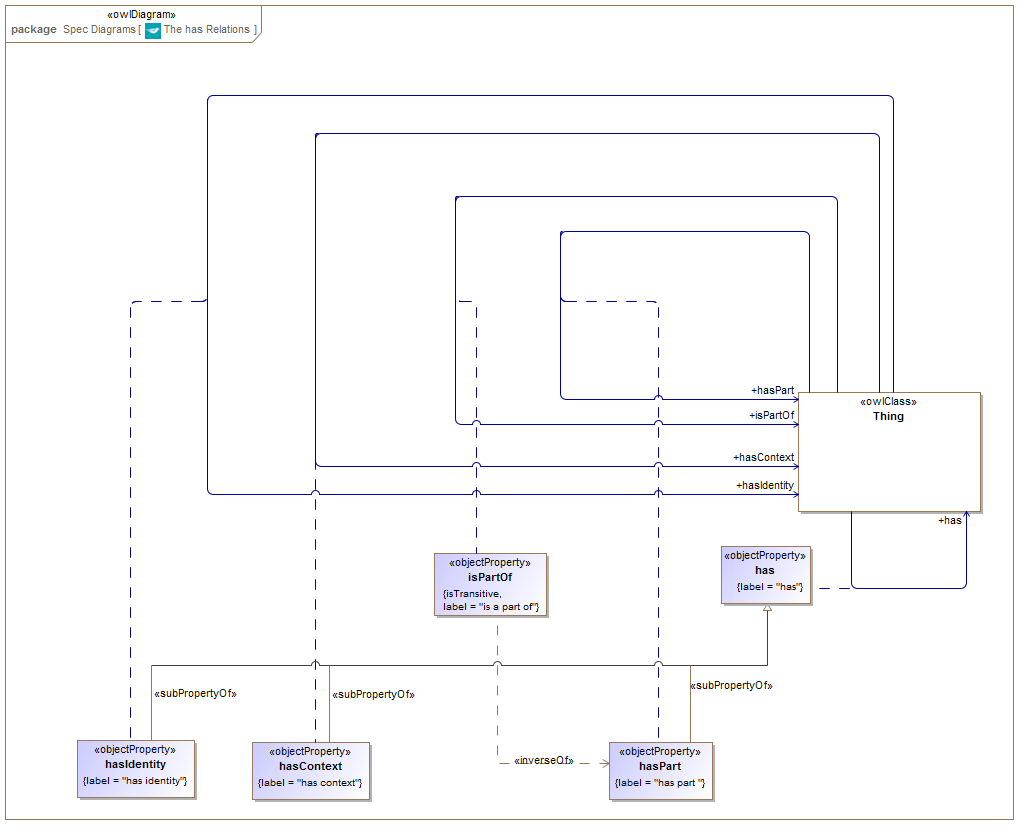
#### Simple Physical Relations



#### Singular Relations



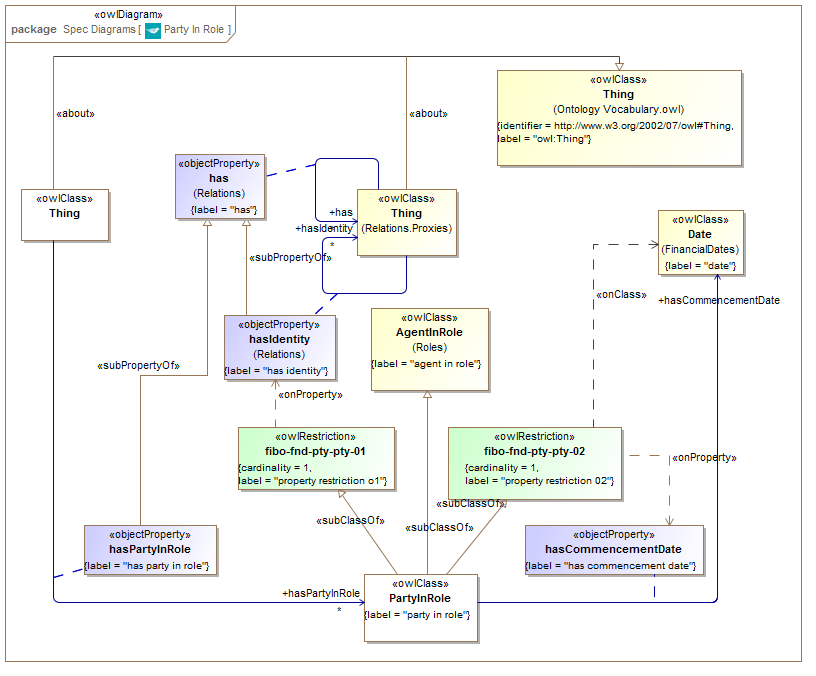
#### The has Relations



## Module: Parties

### Ontology: Parties

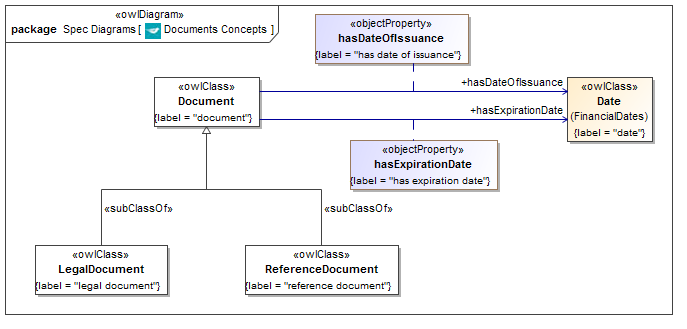
#### PartyInRole diagram



## Module: Arrangements

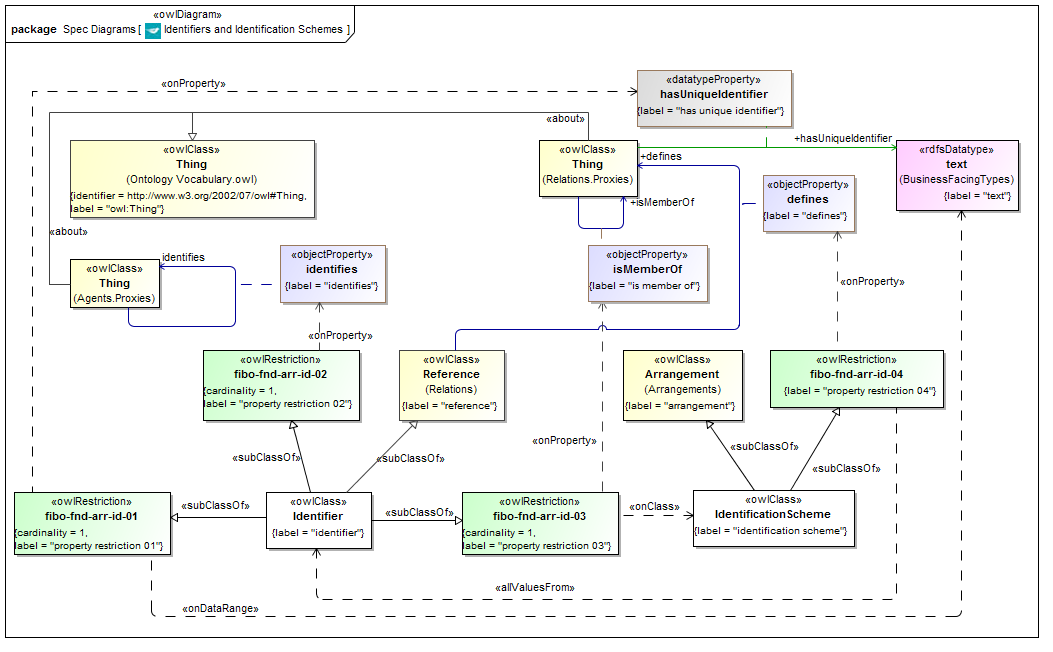
### Ontology: Documents

#### Documents (Change in 64 Financial Dates)

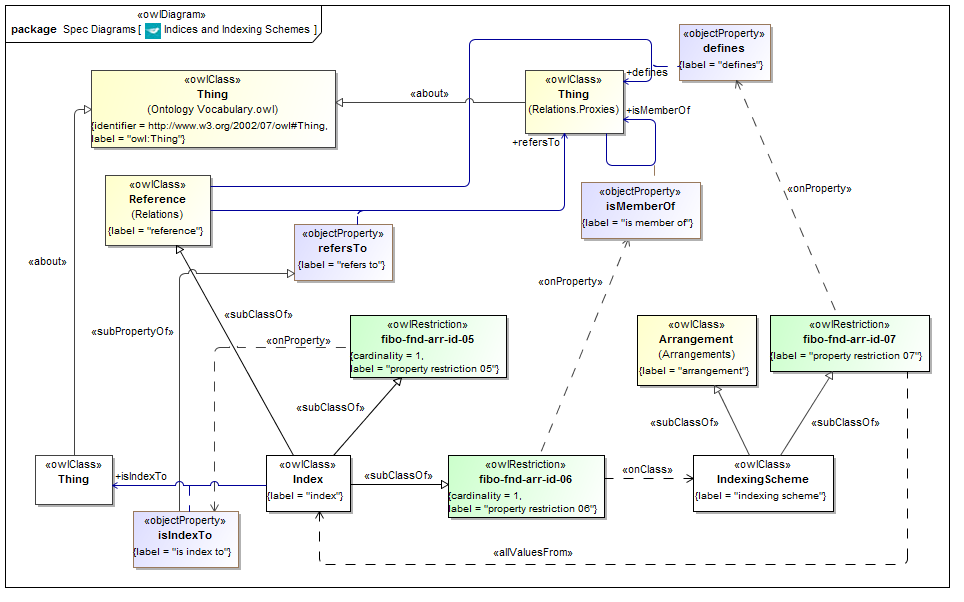


### Ontology: IdentifiersAndIndices

#### Identifiers and Identification Schemes

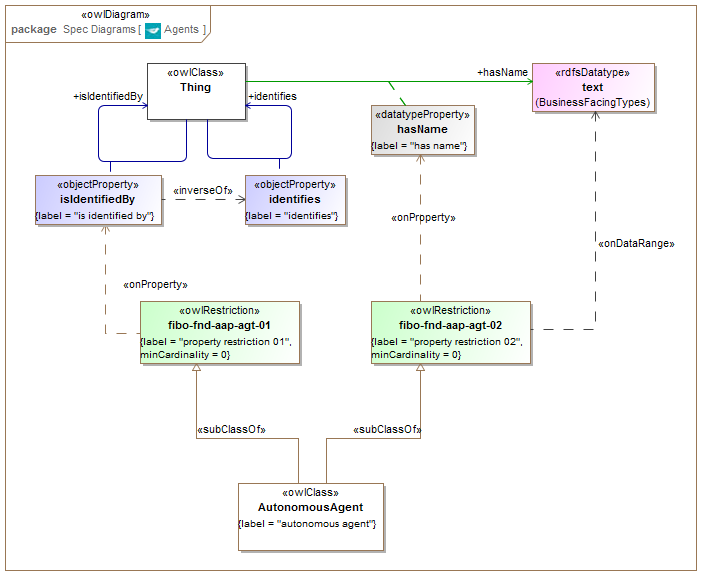


#### Indices and Indexing Schemes



## Module: Agents and People

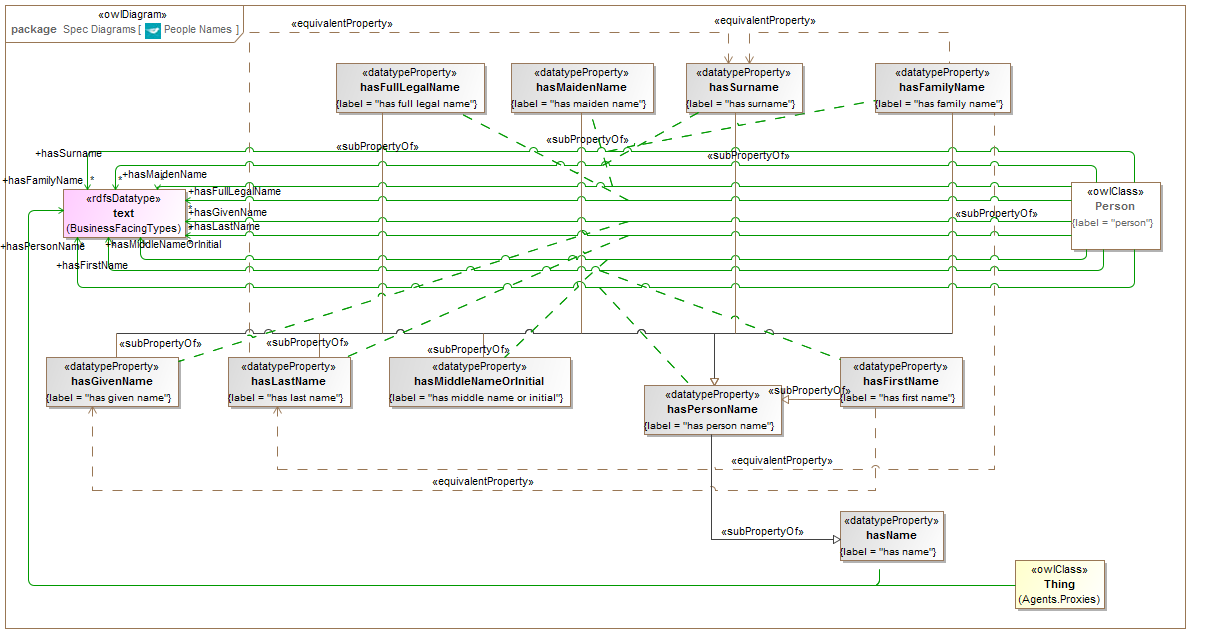
### Ontology: Agents



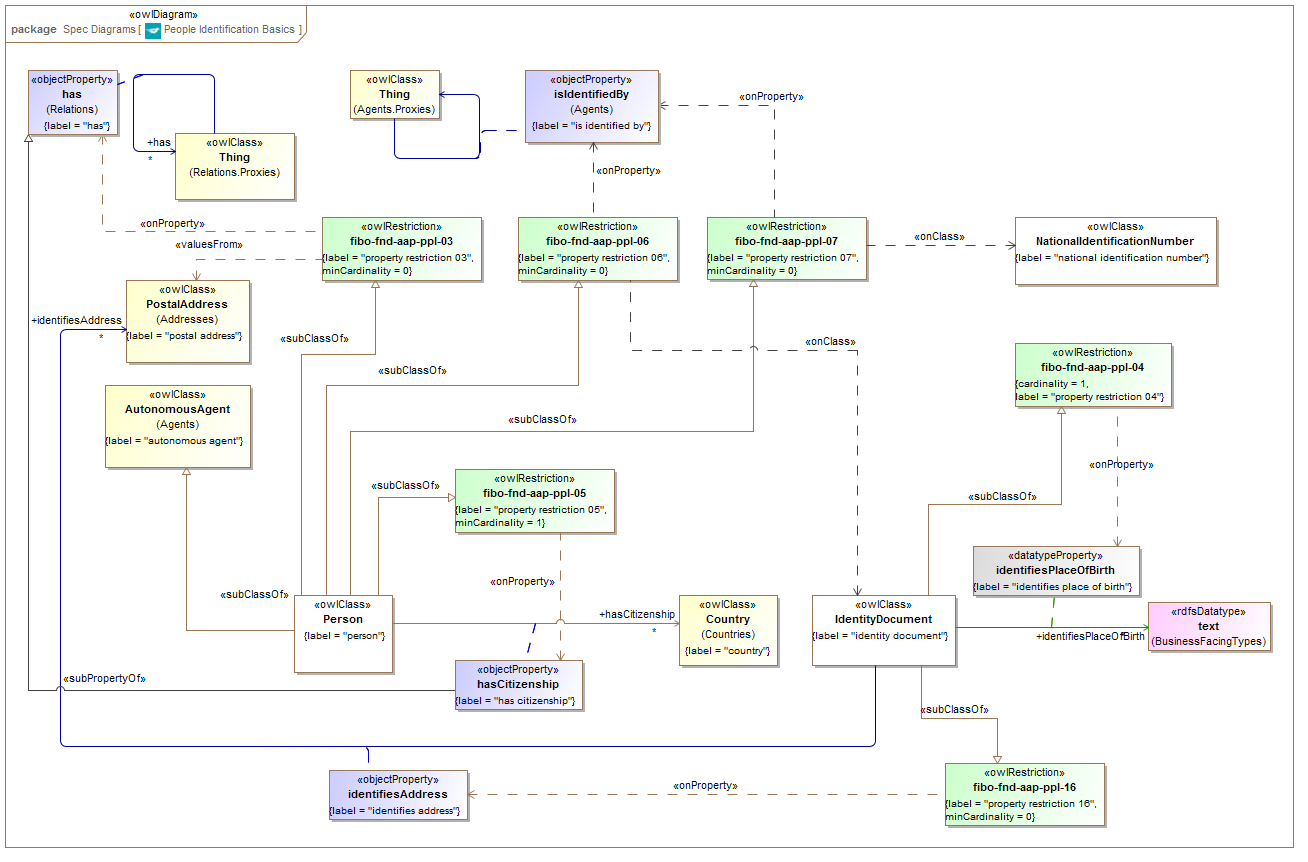
**Changes in tables:** none; property names are the same and packaging is not shown in the tables.

### Ontology: People

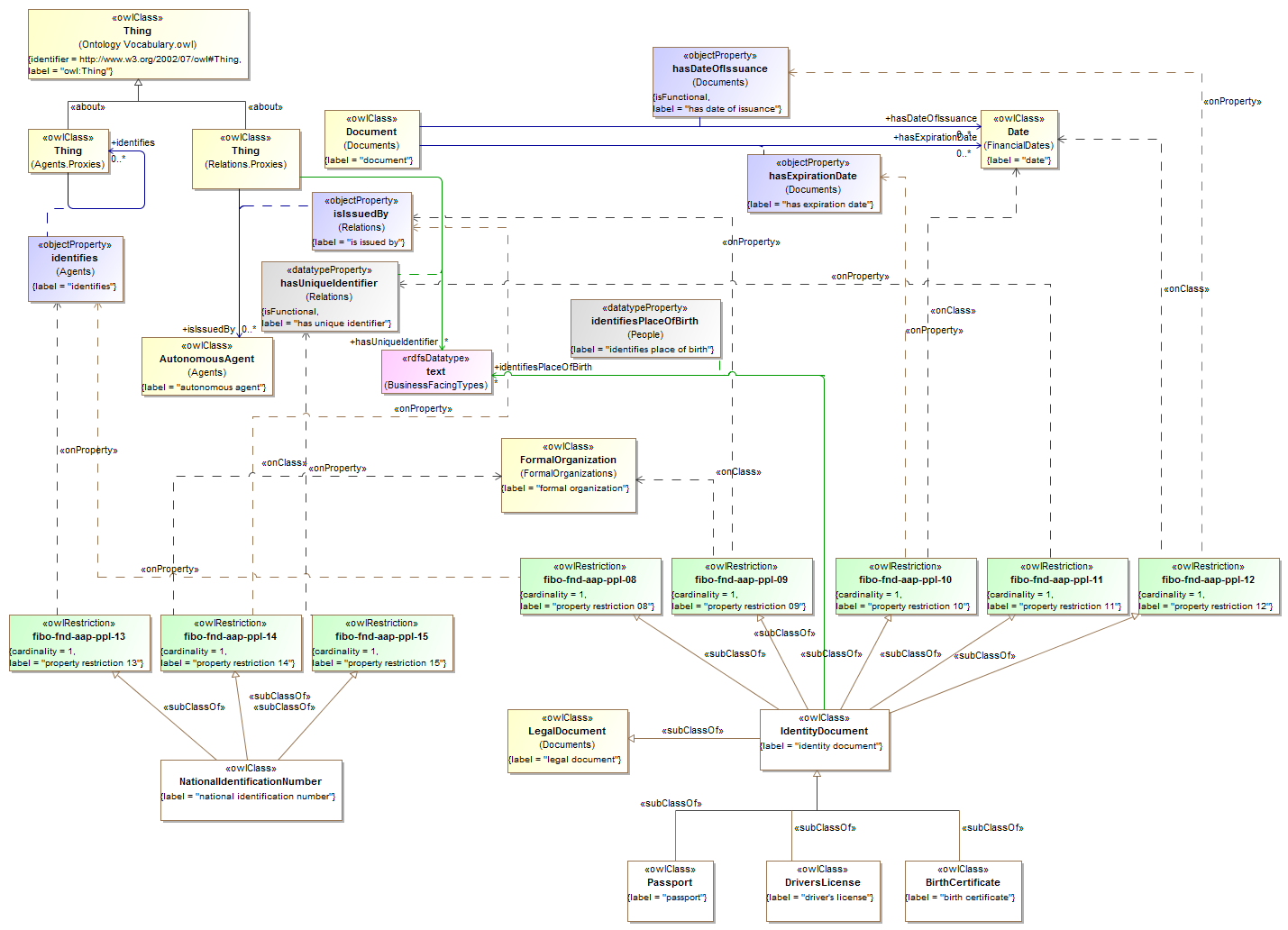
#### People Names



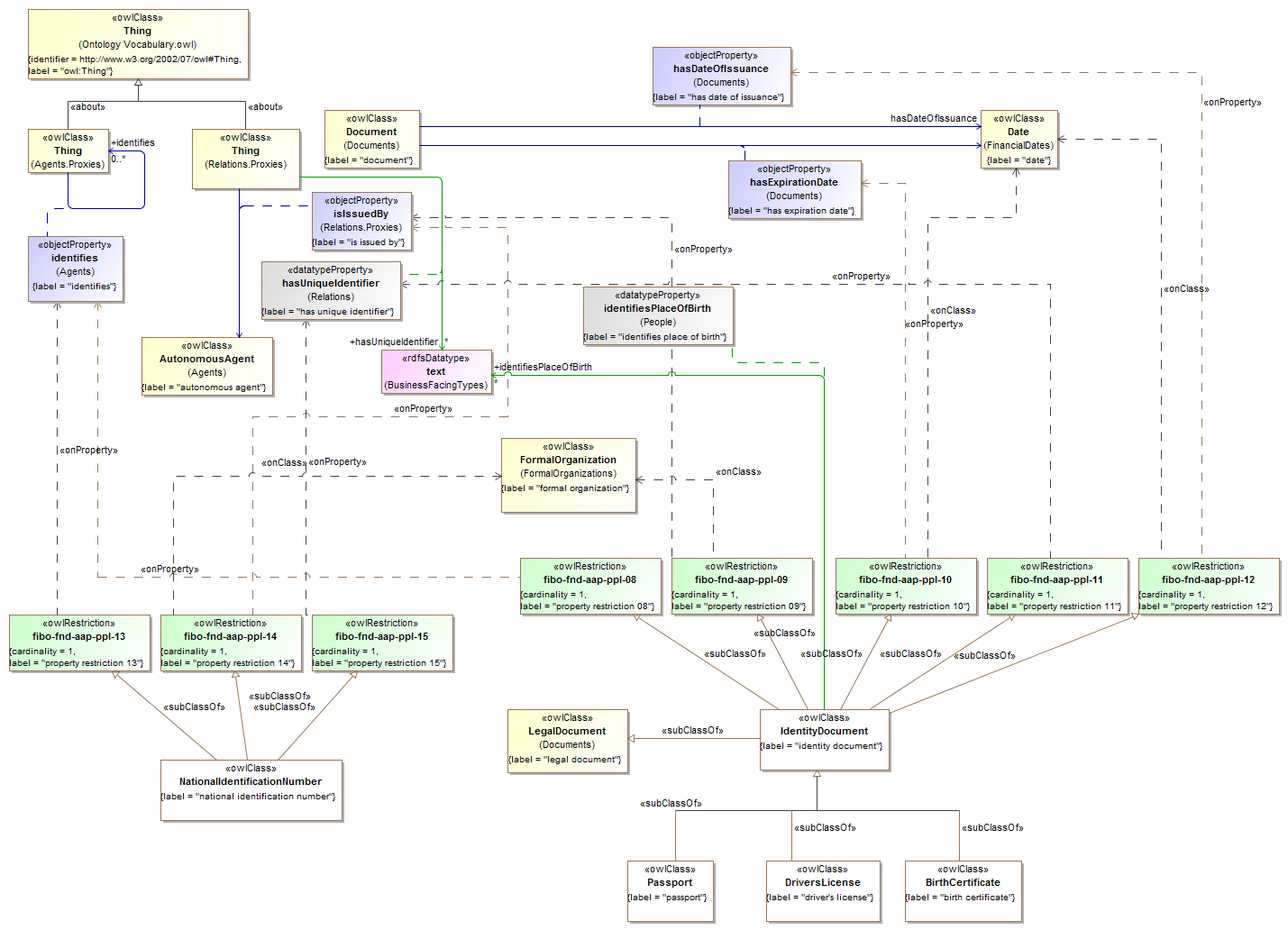
#### People Identification Basics



#### Identification Documents

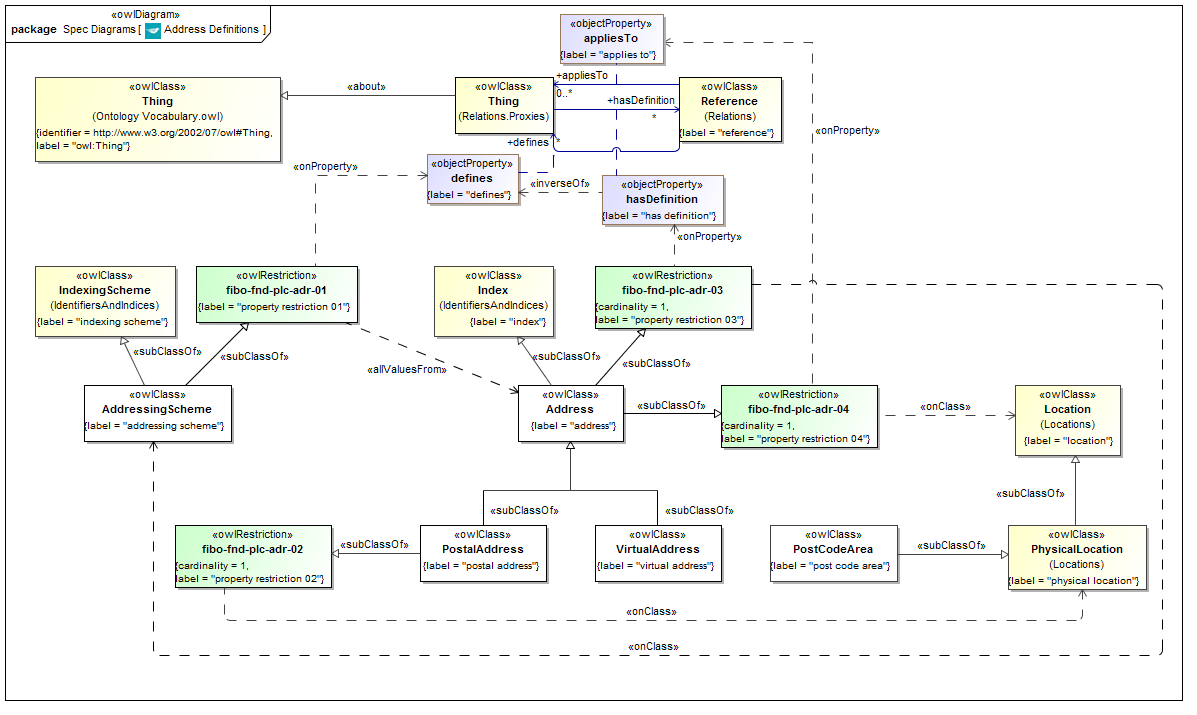


#### People: changes due to Documents change

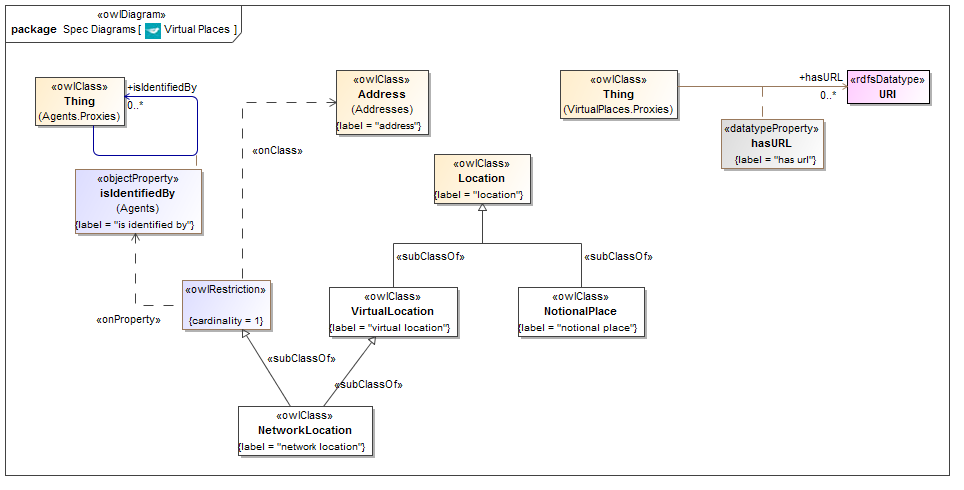


## Module: Places

### Ontology: Addresses

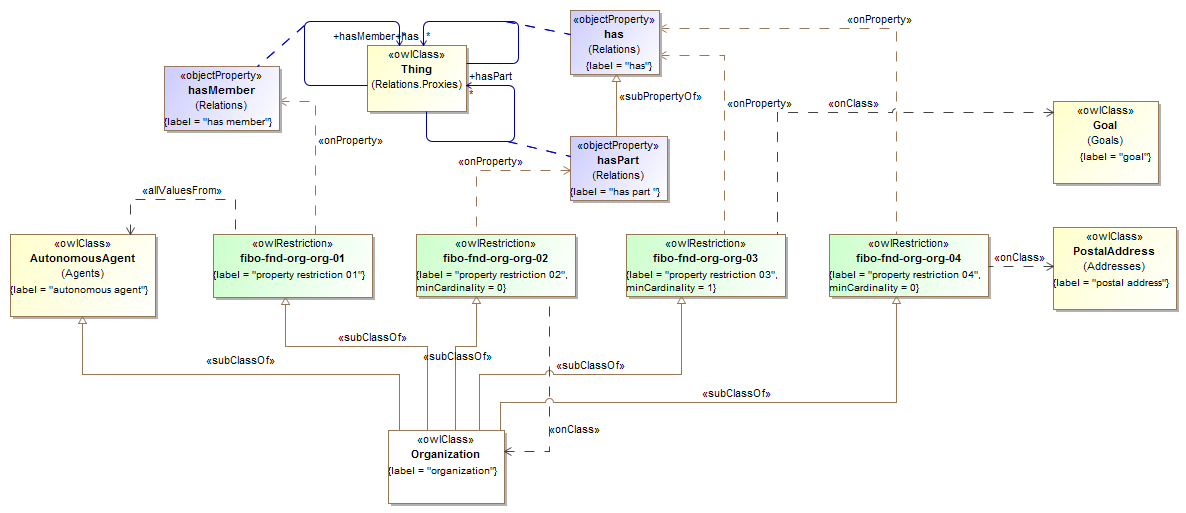


### Ontology: Virtual Places



## Module: Organizations: Cardinality Changes

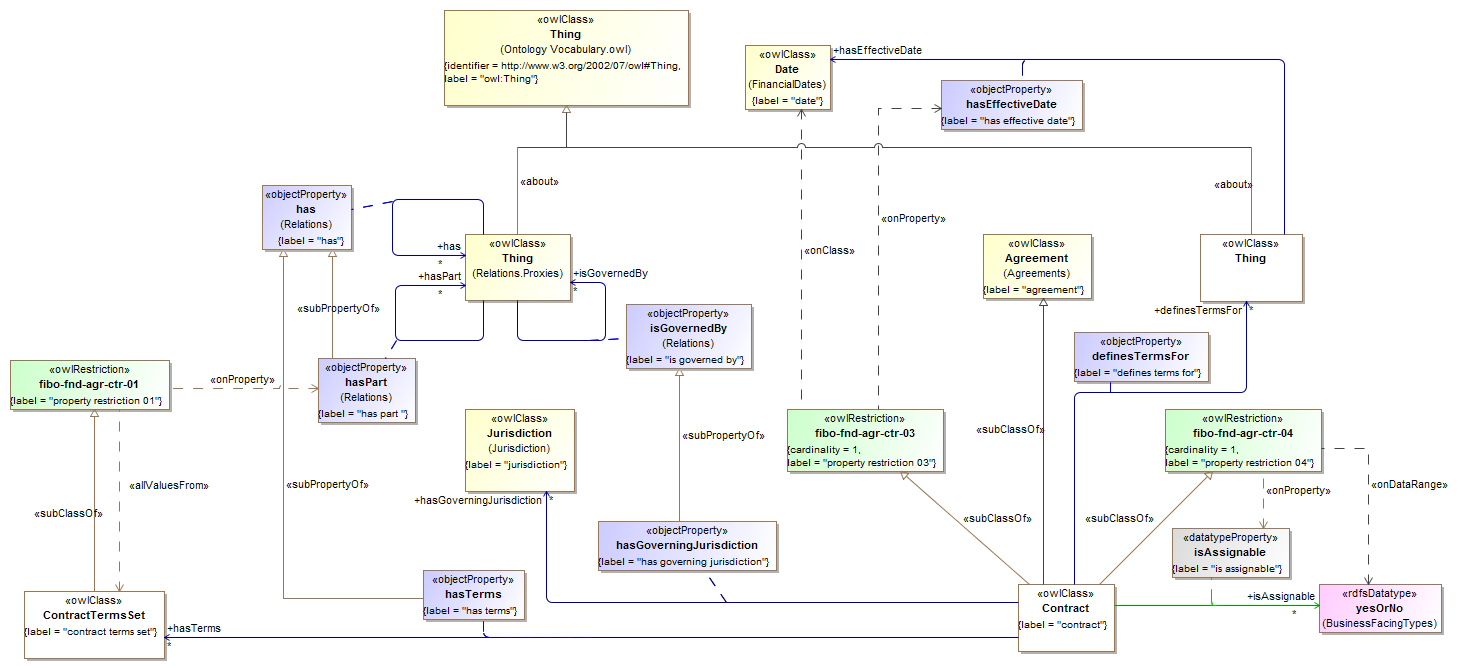
### Ontology: Organization



## Module: Agreements

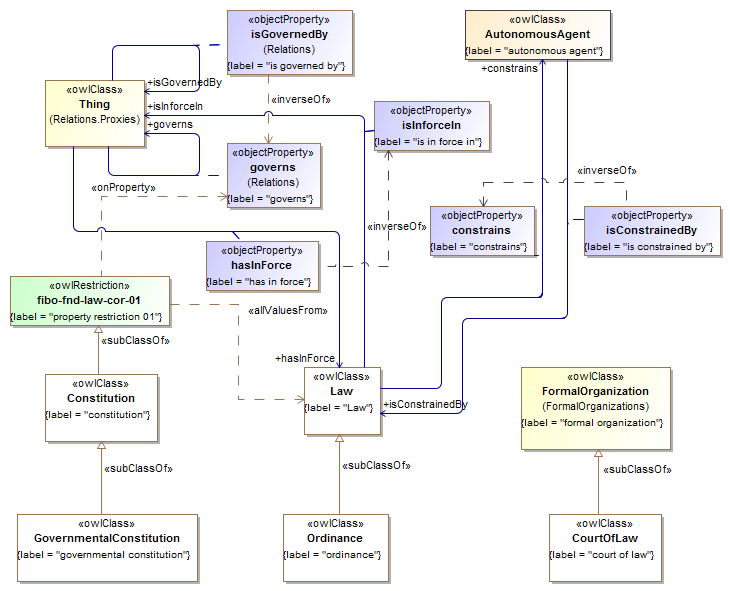
### Ontology: Contracts

#### Contracts Basic Concepts



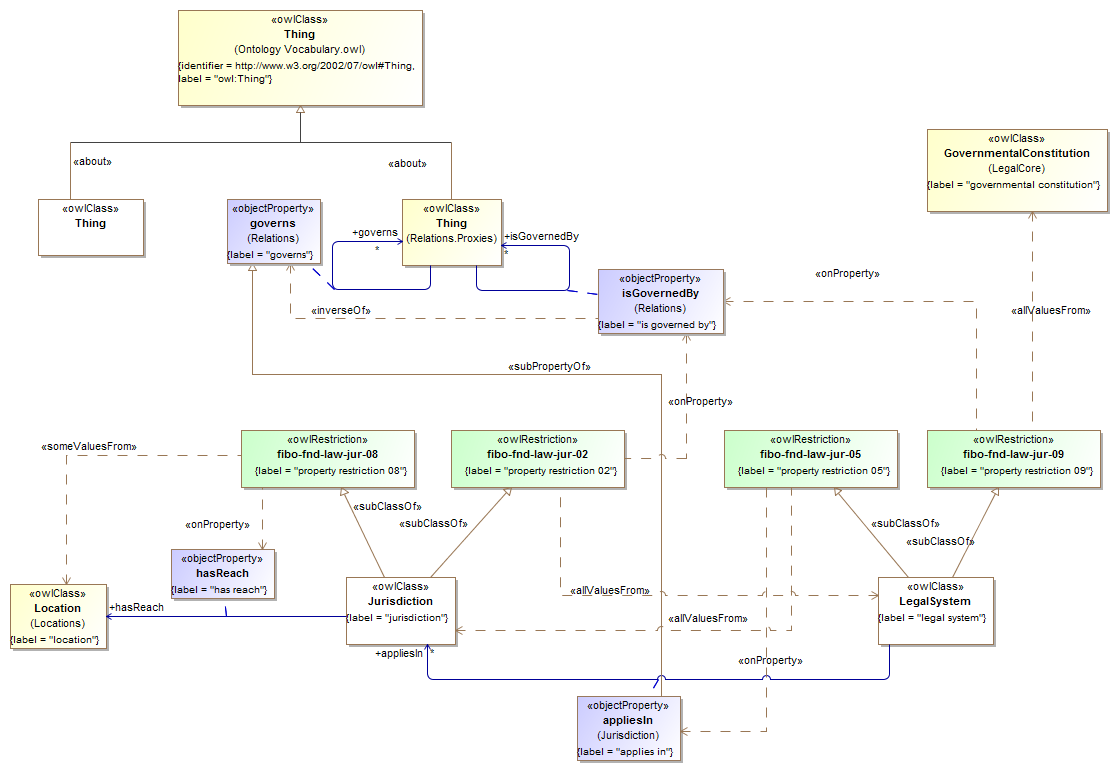
## Module: Law

### Ontology: LegalCore

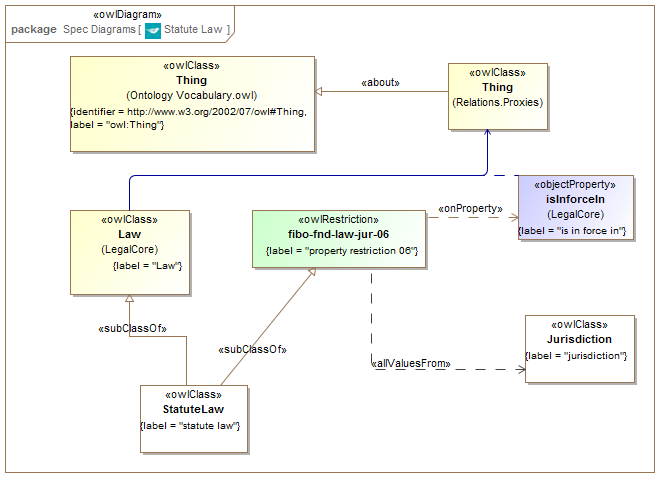


### Ontology: Jurisdiction

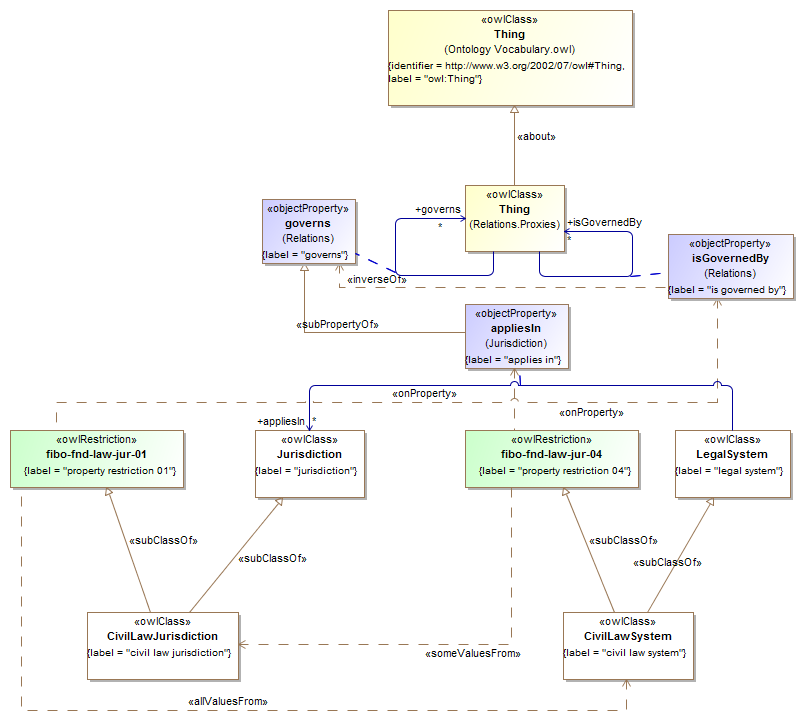
#### Jurisdiction Basics



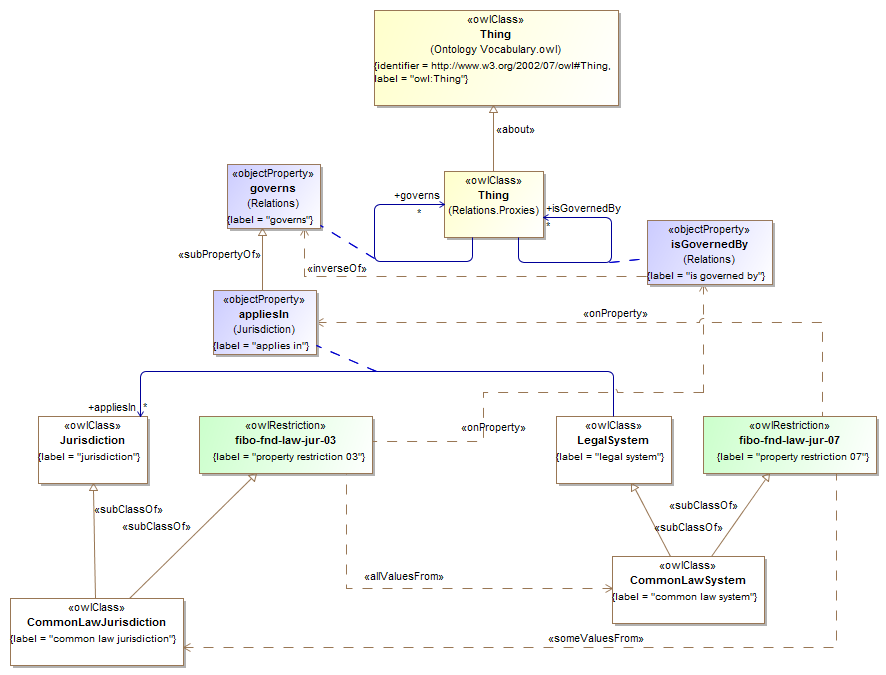
#### Statute Law:



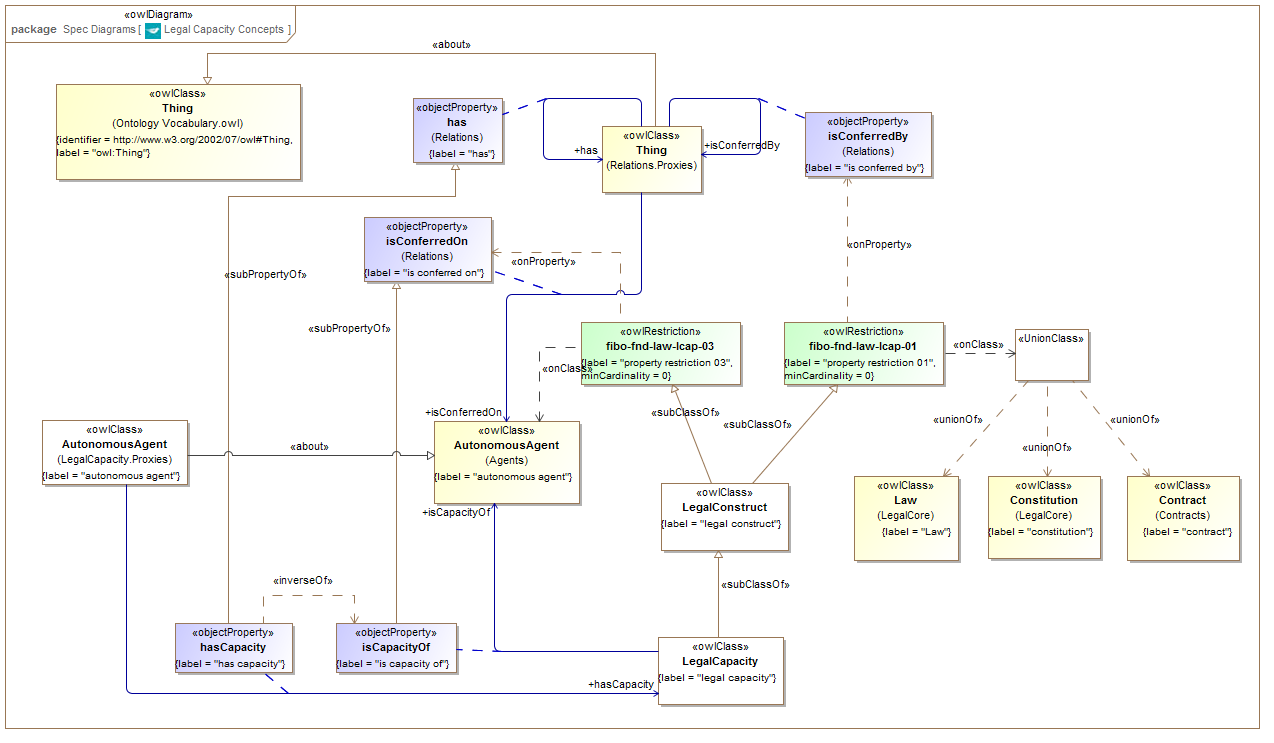
#### Civil Law Jurisdiction



#### Common law Jurisdiction



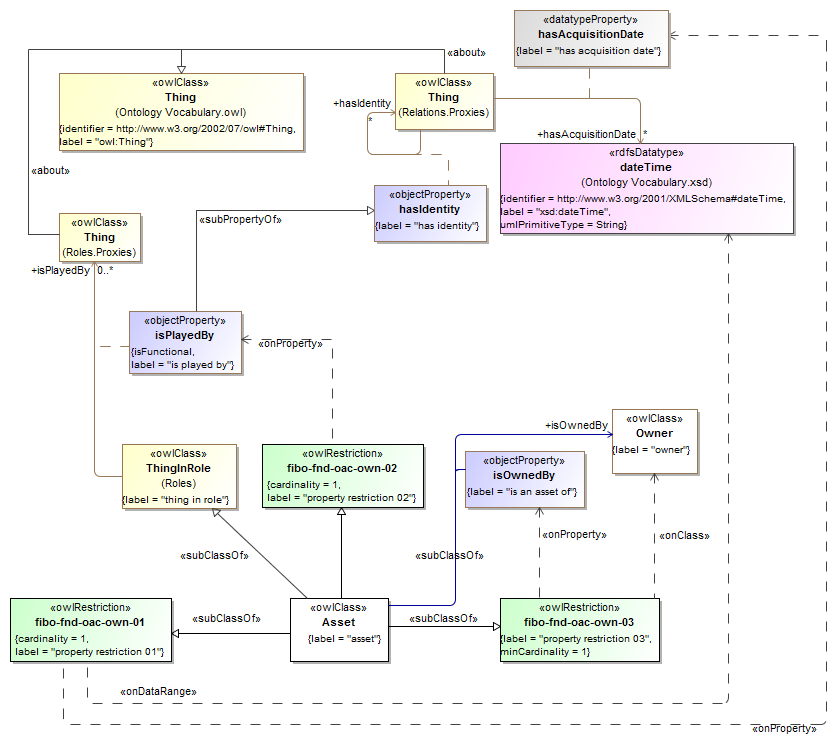
### Ontology: Legal Capacities



## Module: Ownership and Control

### Ontology: Ownership

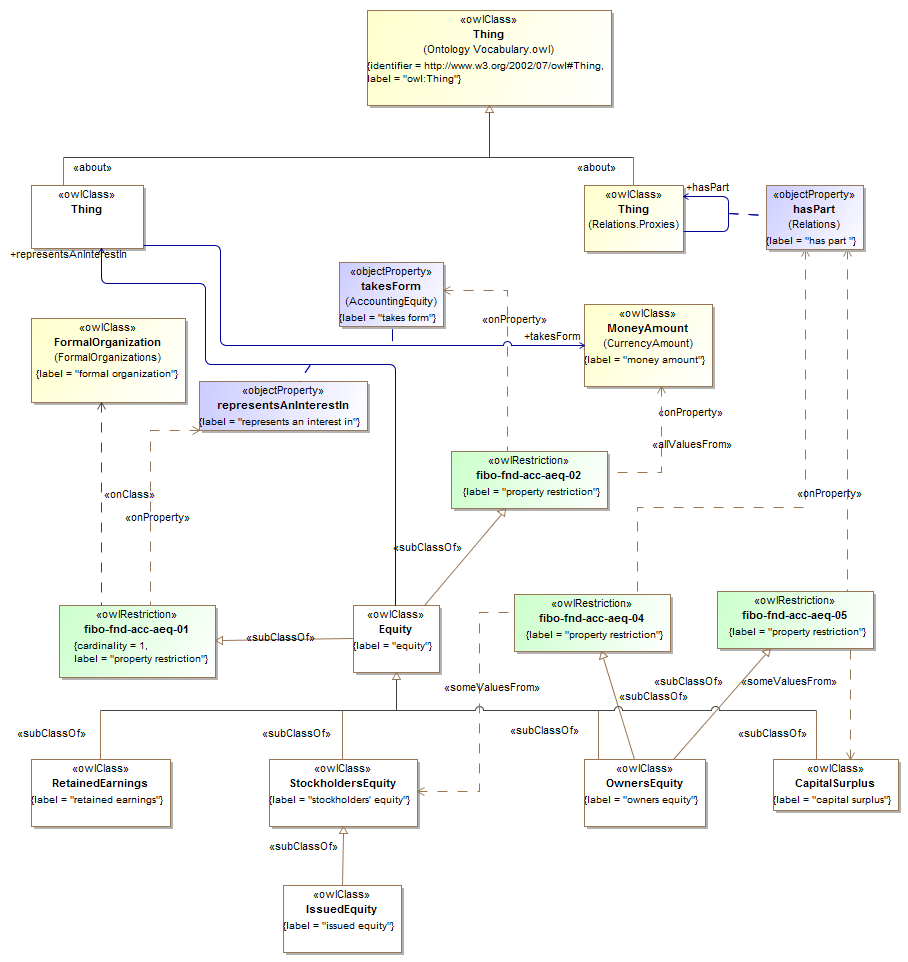
#### Asset Definition diagram



## Module: Accounting

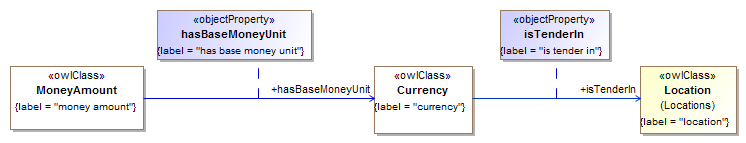
### Ontology: AccountingEquity

#### Equity Concepts



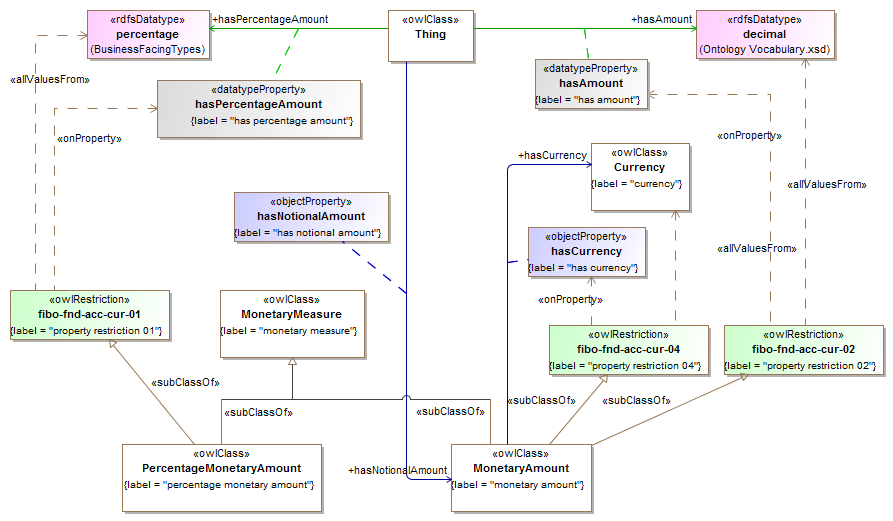
### Ontology: CurrencyAndAmount

#### Currency Concepts



#### Monetary Amounts and Measure Concepts [Issue 10 version]

Updated subsequently for Issue 11



#### Physical Money Amount Concepts

