Workstream 3 Working Notes 13 Oct

REA Terms

Terms Cross Referencing between OWL and UML

Discussed how to cross reference terms. In REA the normative version of the model is a UML model (reference correspondence from Bill McCarthy). Note that this is in a section of an ISO standard ISO 15944-4 and is therefore potentially subject to copyright. Agreed that the copyright issue needs to be looked at, since we want to find a publicly accessible but normative version of the standard from which to take our snapshot. We will defer that discussion to another time since today's agenda is to take REA as an example to document our model cross referencing process for non OWL, specifically UML standards.

Action: Discuss and resolve IP Issues for REA (future activity)

Class to Class Matches: SKOS Match

Put illustrative relationships in for the SKOS Match term linking classes. These are shown as Dependency. However, the model does not yet contain the formal metadata for these, so this is illustrative only, to show where the SKOS Match relationship should go.

We use SKOS Match here instead of OWL Equivalent Class, because the target term is a UML Class not an OWL Class (Thing).

The two classes we are using to illustrate and trial this process are:

- Economic Contract (REA 'Economic Contract')
- Transaction Commitment (REA 'Economic Commitment')

Note that we named the REA contract type as Economic Contract in order to have the same name as the corresponding REA term. This was to distinguish it from other types of contract i.e. all contracts which are not the contract that corresponds to a transaction. We had not done this for the corresponding Commitment type.

Model Change: Renamed Transaction Commitment to Economic Commitment **Action:** Check and rename any other terms that are intended to be identical in meaning to REA terms, to also have the identical primary label.

URIs for UML Terms

Last week we discussed the fact that SKOS Match requires that the items at both ends of the relationship must have a URI. In SKOS terms, the items should both be “Concepts”. Revisited this discussion. UML Classes do not natively have URIs, and last week's conclusion was that the transformation of this model from the UML model repository to RDF / OWL should include the creation of URIs for the UML terms. The conclusion last week was that the provision of these URIs was an

implementation matter and need not concern the model. Discussed this again today.

These URIs for the UML classes may be one of two things: Some wrapper around the UML Globally Unique ID (GUID); or some wrapper around the element name and package name.

Concluded that it would be preferable to have some explicit statement of the URIs for the UML model elements in this repository. This could take the form of a tag added to the Package, implying the URIs of the elements as taking the Package URI plus element name i.e. packageuri#elementname

Noted that in version 2.4 of UML there is an explicit arrangement for namespaces. We are using UML 2.1 and have no intention of upgrading to UML 2.4 at this point. However, we concluded that the way in which we handle these URIs should be compatible with what the arrangement in UML 2.4 will be. That is, the tag shall be called “URI” (all upper case).

Model Action: Add some tagged value to Package, for the packages in which we define UML Snapshots such as for REA. This will be a very minimal profile extending UML to include this URI element.

URIs and Naming

The questions we are looking at for URIs for external ontologies such as REA are the same questions as those we looked at some time ago for our own FIBO model content.

We went back over parts of that earlier conversation for those who were not present for those sessions. Basically, we had agreed that names for elements in FIBO should be not only UML legal but also URI legal. There is an open action to double check on this, but throughout the creation and editing of the existing Semantics Repository, much care was taken to ensure that names were safe against any conceivable restrictions on what characters to include.

However, this thinking won't apply to UML models that represent other standards. Since these are always going to be a snapshot of one model, likely in one UML Package, we would simple have to ensure that the package name is URI legal. Similarly, if the snapshot we take has element names that are not URI safe, we would have to adapt those.

Action: Document this requirement in the process for taking snapshots of external standards in UML. Also in other non OWL formats such as XBRL.

Note that the exception to the above naming, is that our FIBO labels are deliberately and necessarily styled as individual words with spaces. The transformation of these into the legal OWL format of camelCase has been left as a consideration for the transformation.

An open question in those earlier discussions is which of two possible transformation styles to use:

- Replace the spaces with underscores
- Compress the words so that they become camelcase.

There are pros and cons to both approaches, and different people on that earlier call had strongly divergent views about which method to use. We did not resolve that question then or now.

Meanwhile in our current conversation about external standards that are UML: mostly these would be expected to have camelCase names. However, the one we are looking at now, REA, uses the same

style of naming as FIBO, i.e. normal words. I also seems to follow the same convention we have used whereby class names are in 'Sentence Case' while relationship names are in 'lower case'. This means that whatever transformations we apply to the FIBO content will need to be applied to the UML snapshot of REA.

Action: In the documentation for the different Shared Semantics processes, document the above for standards that are provided as UML models, and ensure that the same questions of naming are addressed in other non RDF/OWL based standards.

URIs for OWL Ontology

Also discussed URIs more generally, for the existing OWL content. Having agreed that we should put the URIs in by hand to the existing model, in the relevant tagged value, it was not clear which tagged value that should be. There are several tags in RDFDocument in the existing profile, as follows:

- defaultNamespace
- xmlBase
- namespaceDefinition
- statementForDocument

It clearly isn't the last one, but which of the other three should I put the intended namespace into?

Consensus: It should be replicated in both defaultNamespace and in xmlBase. The namespaceDefinition tag is an optional element which we can ignore. There is also a further element in the ODM 1.1 profile for 'named entity' which may or may not end up being called namedEntity. When that exists, these namespaces should go in that tag also.

Action: Populate the defaultNamespace and xmlBase tagged values in the existing model, with the intended namespace for each package. This will have a root namespace which is yet to be determined.

This action applies both to the Global Terms section and to the FIBO content sections.

Relationship to Relationship Matches

Discussed how to identify matches between relationships.

Note that for the two classes we are using to illustrate and trial this process, Economic Contract and Economic Commitment, the relationship between them has a different label and appears to represent a potentially different meaning:

FIBO: Economic Contract establishes Economic Commitment REA: the association between Economic Contract and Economic Commitment is labeled 'economic bundle'.

In order to establish and document the matching process, we will proceed for now as though the intended relationship has the same meaning.

Discussion points: 1. What relationship should we use between relationships that are intended to have the same meaning? 2. How shall we model this in the model?

Consensus on (1): The relationship should be SKOS Match; this has the same intent and application whether it is between classes or relationships, in that it is a relationship between concepts.

On (2), the issue in the EA modeling tool is that it is not possible to attach a relationship to another relationship. This appears to be a peculiarity of the EA tool, since in MagicDraw it is possible to do this.

The only way to achieve this in the current EA model is to replace all Associations in the target UML model with Association Classes. Agreed to proceed on this basis.

Note also that due to another peculiarity of EA, it is possible for Association Classes to be shown on diagrams with the class element removed. In fact, for any EA model with an Association Class, if another drawing is created with the two classes in question, only the relationship is automatically drawn. This works in our favor, since it becomes possible to replace the UML Associations with UML Association Classes, and still have a diagram with only the Associations, such that this diagram looks exactly the same as the original intentions of the UML model. Illustrated this, and all agree this is the way to proceed.

Conclusion: Use Association Classes for target UML model associations. Use SKOS match to link an OWL Object Property (relationship fact) to a UML Association or other relationship which has the same intended meaning, if there is one.

Open questions: this is as far as we got on this question, but a couple of questions remain to be dealt with on our next session:

- What if the intended relationship is an Aggregation or a composition? (there is no Association Class for these);
- The intended meaning of most UML relationships will equate to a pair of OWL Object Properties, such that the Association Ends, being Properties in UML, each correspond to the intent of one of the two triples in a pair of Object Property relationships, in the event that these Object Properties are 'functional' and one is the inverse of the other. Therefore the true mapping should be between two Object Properties and one Association

This latter point may also help to make sense of the naming difference whereby the REA relationship we chose to illustrate these mappings is called 'economic bundle'. It may be sensible to consider a pair of relationships, where a contract establishes an economic commitment and this commitment implies the existence of a Contract.

Semantic Matches for Attributes

The arrangement we worked out for SKOS matches on relationships won't work for UML Attributes, or for the FIBO ODM OWL Datatype Property construct, since lines can't be drawn between these.

We will discuss this next time as we were out of time for this session.

It may be that there is no solution to this and classes should only be asserted as having the same meaning if all the UML Attributes in the source class become OWL Datatype Properties in the semantically identical OWL Class. SKOS Relationships Summary and Decisions We use SKOS Match where the intended meaning of an OWL construct is the same as that given in the external ontology.

If the meanings we have are intended to be broader or narrower than the terms in some external

ontology we would of course use the corresponding SKOS terms for broader and narrower meanings. However, the intention of the Shared Semantics work is to take terms which we know we need and ground them in the terms in established standards and semantic models, so for the most part we would expect to use SKOS Match, adjusting our model if necessary to have terms from those external standards as a basis for other terms we derive from those in our model.

SKOS terms must have something which is a 'SKOS Concept' at both ends. Therefore terms in UML model snapshots such as the REA one are to be annotation in UML as being SKOS Concept, by way of some additinal UML metadata.

A minimal UML Profile is required, which will have the following two UML extensions, for use in REA and any other standard which is normatively defined in UML not OWL:

- The tag 'URI' added to UML Package
- The tag 'SKOS Concept' added to UML Class and all relationship types, and Association Class
- Additionally, UML Association Classes are used in place of UML Associations.

The adjacent Workstream (Technical Model Framework) this week included a presentation by Elisa Kendall on the recommended Dublin Core and SKOS terms and how these are modeled using ODM.

Attributes will be looked at next week.

Detailed Notes

Diagram: OWL Metamodel and Profile

Namespace

Discussion on where the metadata for OWL Ontologies «owlOntology» should go.

Note on owlOntology

Where it goes right now:

defaultNamespace AND xmlBase

also when we upgrade to the new version of the Profile, also put it in the namedEntity tag. This is called named Something not necessarily namedEntity.

The new ODM 1.1 profile will have an additional element called namedEntity.

Diagram: REA Derivation

We recorded the following detailed notes (summarized above) in the course of our conversation.

These are in the order in which they were recorded.

Notes:

Mapping of relationships

UML Association has an existence just as much as UML Class.

IRI Mechanism

Do we have a mechanism to determine the IRI?

This is mixed up with the whole business of how we generate the RDF.

In UML 2.4 the Package has a URI. However we are on UML 2.1.

Possible approaches:

Add this to the Profile and have a tag. But that's obnoxious as it requires lots of work

The URI for each element is derived from the Package. This would require some code to inspect the model to calculate the IRI for each element from the Package it is on.

Other challenges: Element names that are not legal in IRIs.

Policy: Naming is to be compliant with IRI as well as UML.

Approaches: Squeeze out the names and have camel case Replace spaces with underscores.

Where this Question Applies

Two areas where namespace questions apply:

1. the FIBO content - how IRIs are applied to the model elements.

2. the external ontologies 2a: which are OWL 2b which are UML.

What we concluded: Each package has tags defining the URIs Defined whether each package is an ontology or not. If it an ontology what its URI is.

Need to be compatible with UML 2.4.

Decision:

Q: Where to put the URI in the UML?

A: Have a URI that identifies this as a local snapshot of this.

For ISO Standards there is not a URL since these are pay per view. this also means this is copyright material by ISO.

On the Where question: Anticipate UML 2.4 and put a URI Property on the package. What is the name of that tagged value? Answer: “URI” all upper case.

This applies only in UML Package. The elements will then be /packageURI#classname

Overlapping Concepts:

SKOS Match - are we using that as an absolute functional equivalence and if so is that going to be adequate. What about overlapping concepts.

What about SKOS Match at the property. Do not use dependency for SKOS match.

To be addressed at our next session.

Other questions:

SKOS Standard - it appears that the items at either end have to be of the type SKOS Concept.

There are implications to this. Any items that are stereotyped owlClass that are linked by a SKOS Match will have to have an RDF Type statement saying that they are SKOS Concept as well.

Is this really a problem or is it not inferred anyway? Opinions differ at present on this. Have to make some decisions about what is legal on the generated RDF .

Possible ways round is to make all of the SKOS Properties annotations. We have not done this. At present some SKOS terms are annotation but some e.g. matching properties are not. So somewhere up in the hierarchy we may or may not need to do something to say explicitly something is a SKOS concept.

If they are (OWL) Annotations then this won't be inferred.

We could achieve some of the same ends that we get through inferencing through SPARQL Constructs.

Is this an RDF Ontology with an OWL ontology embedded in it? If so, how do we extract the underlying OWL ontology and what sort of reasoning regime we will have around.

From:
https://www.omgwiki.org/OMG-FDTF/ - **Financial Services DTF wiki**

Permanent link:
https://www.omgwiki.org/OMG-FDTF/doku.php?id=ws3_thursday_13_october&rev=1318941714

Last update: **2011/10/18 08:41**