FIBO v2 Specification Automation Call

*15 March 2019*

# Attendees

* Juergen
* Cory Casanave
* Pete Rivett
* Bobbin Teegarden
* Jason McC. Smith
* Jim Logan
* Mike Bennett
* Larry Johnson

# Agenda

* JIRA Co-ordination
* LaTeX Doc Creation
* CCM Diagrams
* Also URIs per PR email

**Note:** For each topic in the notes below, we start with Pete Rivett’s email notes (in green) followed by the meeting’s discussion and conclusions on that topic.

## Actions and Follow-ups

Technical content tomorrow can continue as workstreams

Do we need a Steering meeting next week?

Do the tech streams need to meet next week?

* More ad hoc for the working sub groups

This steering call again in 2 weeks. Same time as this, every 2 weeks.

# JIRA Co-ordination

1. **JIRA coordination**

For **cloning of Issues,** JIRA has a myriad of options, including bulk and issue-at-a-time.

See for example <https://community.atlassian.com/t5/Jira-questions/How-to-clone-issues-to-a-new-project/qaq-p/14885> and

<https://bobswift.atlassian.net/wiki/spaces/JCPP/overview>   (commercial plug-in depending on number of users licensed)

<https://community.atlassian.com/t5/Answers-Developer-Questions/copy-link-issues-between-projects/qaq-p/467787>

JIRA also has a nice API accessible from many languages.

So I think the next step is to work out what we (**all**) want and the process. For example:

- (after an initial bulk copy) do we want it bulk, event-driven or user-initiated?

- If not user-initiated how would we signal (via some sort of Workflow Status or Tag?) in EDMC JIRA which issues should be cloned?

- How do we keep the issues linked?

- Do we want any post-clone changes/comments to be copied over too?

- If EDMC works with discipline (one issue - one change - one pull request) then each (closed) JIRA issue should be associated, from JIRA, with a GitHub pull request which gives us a set of diffs for the complete changes to the machine readable OWL. Or, alternatively, a “patch” that can be run against the old OWL to produce the new OWL. We’d want this available on the OMG issue (BTW for the LCC 1.1 RTF we just completed, I attached OWL-change patches manually to each OMG issue from the GitHub source).

## Meeting Notes on Jira

Larry spoke to Mariano about this part.

Relation to documentation and continuity on this. MB says the code is the documentation.

If we will have something installed that is interacting with OMG data, there has to be a maintenance plan. Technology debt.

Example of technology debt – the process we followed to get Agenda automation from PoC to a production arrangement with documentation on how to use it. LMS to add documentation on how the code operates. We would need to do this with the JIRA is also documented.

See also Data Residency issues.

## Requirement

There are many JIRAs we DON’T want cloned:

* Only things for Release content

Process for getting things over?

At what stage in the lifecycle

* When raised?
* When resolved in EDMC JIRA

# Process

Following on from the Jira discussion

### Initial Process Comments

Where the process is executed?

What state is it in when it comes across?

* Both when it moves or when it does not move?

EDMC JIRA Lifecycle (moves through the states given in JIRA)

Change made

Pull Request – is named after one issue

One branch, one GitHub pull request, one issue

Current: Peer Review

Pull Req is merged (commit) – updates the view in the JIRA system.

Leverage this in OMG. As used in LCC. Includes detailed patch from GitHub alongside change details. Want to re-use that.

Process – when to bring it across.

e.g. clone it to OMG JIRA once the change is committed, along with the detailed code changes. Then the \*TF reviews it.

Who reviews where?

FIBO OMG \*TF – only minor differences but can be different people.

Then the \*TF does a formal ballot on that.

EDMC issues are the same as OMG issues but only a subset of EDMC issues are cloned to become OMG issues (we propose).

OMG Jira will show a set of issues with detailed Git change proposal, and normal balloting. Addition is that the is a link on the OMG Jira back to the EDMC Jira.

**Proposal:** Issue comes to OMG once resolved on the EDMC side.

**Consensus:** Yes.

Event driven or batched? By event, so we want some automation or manual at least for now with a person hitting a button to bring it over.

Event = Issue being closed on the EDMC side.

Actioning the Pull Request – does not automatically close the JIRA though this can be done.

Therefore: Event = the Jira being closed on the EDMC side (not any GitHub action as such).

A complexity: If the Issue and the Pull Request impacts multiple ontologies. Come back to that scenario. This is for hybrid of Release and Provisional.

Maybe any changes that are not Release will vanish in the Redline assuming that that is generated from the Release metadata tag.

### GitHubs at OMG

Expecting a GitHub Repo at OMG in the near future. Mariano on that.

For this process: Do we rely on the EDMC GitHub diffs, or do a Pull Req into the OMG GitHub and do a diff on that.

LJ: Not seeing function points for what we are trying to do.

What data are being exposed outside of OMG that should ont be

What is the state of things coming from EDMC?

If data reside outside OMG direct control, per Data Residency, should still be under contractual control. Do we have a QoS agreement, conditions, storage requirements, DP expectations etc. for the state when something transitions over to OMG?

Answers: No-one is proposing that EDMC is the storage for OMG content. Hence suggestion of OMG having its own GitHub. If not, would still be a diff applied, from the EDMC GitHub, as part of raising the issue in the OMG Jira. This has all proposed changes, is human readable and can be applied to existing OMG file.

### Decisions on EDMC end:

EDMC Membership very broad, including some OMG members; EDMC is an OMG member.

OMG has member: EDMC. This raises an issue in the OMG Jira and that is balloted in the normal way. The difference is only that the input to the OMG Jira is via an API rather than a person. Anyone looking at OMG Jira sees the usual things.

### OMG GitHub Repository Requirement

LaTeX and OWL or just LaTeX?

* We did not yet answer this
* When available at OMG?
* Also see URL question.

Would we put the new version of the (OMG) OWL into the OMG Jira i.e. apply the change apply the OMG publication transformation (for separate OMG URLs)

Even if we did not have the OMG Git we can apply ones from EDMC.

When are we likely to see ‘a’ GitHub from OMG? What Repo.

* Each group would have its own private repo

# LaTeX Doc Creation

1. **Specification generation**

For **Annex text production**

(**Jason**) What format does the LaTeX need for the OWL output, in order to generate the sort of table-strewn text that the FIBO specification documents utilize (see finance/18-08-08 for an example)?

**Diagram inclusion**

How would diagrams get pulled into the right place – could we generate a reference to SVG files that the LaTeX publisher will pull in?

## Specification Generation Comments

If we are looking to generate for models, what will they look like both for text and tables.

JMS: within MagicDraw. Both for model and documentation snippets.

Uses the MagicDraw reporting thing. Includes diagrams.

Existing spec: table structure is trivial in LaTeX. Uses simple XST stylesheet to generate LaTeX. Given the right mappings from RDF a similar script could easily be create.

By whom? PR can do this. Can possibly add something in the EDMC Publish process to run this once it exists? Is that sensible?

PR we have 1 issue that corresponds to a diff in GitHub. Given that, we have current state OWL, the current doc generated from that. Apply the diffs (new OWL); generate new version of Doc. Apply the new Overleaf differencing capability to give the Change Bar doc to show the changes for that one issue only. Can that Change Bar doc be incrementally updated over time so it becomes input cumulatively so that the final one has the annotated change bars for all the issue numbers.

JMS: Yes. Makes use of the Git repo.

So we have the end result of the RTF for the RTF Report.

Can this also be used to assist voting at \*TF on the issue? Generate the revised text in Jira based on stuff put in the edits. Meaning? Can we generate a change bar with only edits from one Jira, so it is presented for reviewers for ballot.

PR: present complete list of changes, so e.g. ‘on page 5 replace this with that’ where that is autogenerated from the model.

JMS – maybe. Flips around from what we were looking at before. Ensure things are in synch. The regenerated doc remains a branch. Shows the change but is not merged until it is voted on.

### Also

Share tools that we use for this.

### Editor

LaTeX Editor: JMS uses ‘Overleaf’ on line editor.

Has a decent rich text view. WYSIWYG

Purpose:

1. Generation of Redline document
2. Generate the actual document

JMS has developed a diffing system that will take any 2 versions of a LaTeX document and redline any changes between the two.

Issue: Tying that back to JIRA to identify specific JIRA issues against specific document changes.

* That piece to be created still
* Push-button operation
* Overleaf editor is extensible – I can identify which change is associated with a given issue FND-123 etc.; will do a diff, find the changes, and retain that information for when the entire Redline doc is generated

None of these is a blocker.

Needs to get info from JIRA, tie in a YRL from somewhere. JMS sees no issues in that.

If FIBO becomes an ISO standard (via the path process)? OMG can say it will host the documents, people can use the editor, generate the redlines. Easy for users. OMG provides the tool.

We think ISO have in-hose editors but most of the stuff is done at this end via the fast-track process.

Once it is in a form where you have control of the structure, and given that the template is already per ISO standard, minor changes at the end should be OK.

# CCM Diagrams

1. **Diagram generation**

(**Mike**) can we make the diagrams more predictable so that we know in advance (when generating the above text) what diagrams will be present so that the text can contain the diagram reference in the right place, even if (for now) the diagram has to be hand-curated and named? For example two diagrams per ontology and one per class?

And (**Jim**) can we run a bulk diagram export to generate all diagrams as SVG into the required location? Though see pragmatics in the next paragraph as to why this might not be a good idea.

(**Jason**/**Manfred**) Given that we’d like to have a new (transient) version of the specification for each issue (yes really, so that the LaTeX system can do the automated tagging of the differences with the issue number), how long pragmatically will all this take, to generate a new version of the spec for an issue, and diff it with the previous version? And can the LaTeX system produce a concise set of specification diffs for issue balloting purposes (so that the transient version really is transient)?

Part of this is that we need to be reassured (**Jim)** that regenerating an unchanged diagram does not result in any detectable differences (which might happen, for example, if the date of generation were somehow included).

Otherwise we’d need to determine which diagrams have been affected by OWL changes for a given issue and only regenerate those.

## Meeting Notes on Diagrams

Relevant for the document as well.

Proposal is for precise diagram naming.

Text will have an ‘include’ for the diagrams.

e.g. Corporations/StockCorporation – include Corporations/StockCorporation.svg at a stated point in the text. LaTeX expects to find that in the appropriate location.

Does this work? JMS generating the diagrams from MagicDraw. Includes the LaTeX snippet to include the diagram, which knows the diagram name.

Can we generate a CCM report with only diagrams and their snippets, and clone that to the other LaTeX thing generated from the RDF and Stylesheet. Yes.

Possible strategies: Package name + whatever.

### Possibilities:

* One per ontology (too big)
* One per class
* One report from CCM not useful. Easier you save just the diagrams.

How to then tell whether a diagram has changed?

LaTeX diff only looks at the text

Store the SVG files in the Git Repo and use versioning and look there for changes.

That’s a new enhancement – does not exist but yes we need that.

Then what does the doc have e.g. pre and post or just post. Which do we want to see?

PR: Pre and post is best.

# URIs

1. **URIs**

One item which we didn’t spend enough time on IMO, is the fact that we currently publish and identify each (released) FIBO ontology with 2 different URIs. For example:

<https://spec.edmcouncil.org/fibo/ontology/BE/Corporations/Corporations/> and

<https://www.omg.org/spec/EDMC-FIBO/BE/Corporations/Corporations/>

Are we happy with that? Let’s think beyond the pride of organization branding and control and ask the more important question **is it good for our user community?**

Will anyone seriously write FIBO applications using OMG URIs instead of the EDMC ones?

As a precedent I see that our Retail TF is retaining non-OMG URIs for their established specs.

I’m curious about ISO – for OMG standards such as UML adopted by ISO through the Publicly Available Specification route (the equivalent of OMG’s RFC route used by FIBO) does ISO require a set of machine readable files with ISO.org URIs? I believe not. Why then does OMG (seemingly) insist on such a change? Why can we not retain the original EDMC URIs?

Even assuming we retain both, it occurs to me that the bulk of the ontologies themselves do not use the full URIs but namespace prefixes and/or XML entities.

For example, the following declaration of a class within the above ontology is identical for EDMC and OMG.

              <owl:Class rdf:about="&fibo-be-corp-corp;JointStockCompany">

If we make more disciplined# use of prefixes and entities then it seems the only differences need be the headers in each ontology that decalare the prefixes and entities. That means that any diffs or patches for OWL changes could be totally transportable between EDMC and OMG.

# at the moment, for no reason I’m aware of, we don’t use them in the rdf:about for the ontology itself, the versionIRI (which is automated anyway), some metadata such as “sm:dependsOn” and owl:imports triples.

## Outcomes

* Is EDM happy with changing this?
* Is OMG happy with e.g. using only the EDMC ones?

This is a question for the OMG AB.

Similar to when OMG submits to ISO through the PAS process, which ISO does not require to be changed, so this is a good precedent to follow the EDMC URIs.

Previously, mantra was to use OMG URIs. For FIBO, these exist for purely historical reasons.

JG add to AB Agenda.

Continuity – e.g. if the EDMC were to dissolve.

If the organization ceased to exist someone would still own the URL – e.g. potentially could take on the URIs from EDMC. Also FYN but is not necessarily.

Also make sure diagrams never include URIs - this is a formal requirement.