**DDI Sprint #3
Vancouver, BC, Canada**

**Tuesday, March 25, 2014**

**Minutes**

Sprint participants:

Therese Lalor (Project Manager and Sprintmeister)

Jay Greenfield, Arofan Gregory, Larry Hoyle, Jon Johnson, Alan Li, Ron Nakao, Wendy Thomas, Joachim Wackerow, Mary Vardigan

**DDI 4 Documentation**

It was pointed out that when we model something mapped to GSIM and 3.2, we need to document this immediately and do the documentation in a formalized technical way. We did this for the RDF Discovery vocabulary with SPARQL queries and xpath statements. This is clear and a program could take advantage of this. Even if the mapping is not one to one, it should be documented. Rules for documentation like this should be communicated to groups working on the modeling. We should explore possibilities for how to do this in a formalized way this week.

The import of 3.2 to Drupal will ideally happen during the week, but it depends on the availability of the Drupal developer.

**Packaging**

The group has talked about switching to views and we have a temporary version of the syntax to build a view. This would permit groups to build a master list of all the objects related to the view. If we use XSLT to produce outputs, we need to use XML for views. Achim suggested XMI but it is very specific and complicated. We need agreement on this during the week so the working groups can use this.

If we want something like clickable GSIM, we need to factor this into the architecture for users to create views.

Jay made a presentation on “Toward Packages.”

In this presentation, there is an Upper Level Model proposed based on Basic Formal Ontology (BFO). There are two sides to it: Continuants (things, artifacts) and Occurrents (process and events).

Continuants Packages (abstract classes) include:

* Construct – Concept, representation, class, CDE
* Measure – Self-report, observation, assay, question
* Result -- Variable, report of findings
* Record – Dataset, Index, metadata tag, other annotations
* Presentation – Tables, Model, other analyses

Occurrents are the realizations of these things – Process and Events.

These classes have a provenance from another ontology.

We could put this together in a metamodel with triple tags and index them. Triple tags have a name space, a key and a value. The advantage of this approach is that you can begin to specify what is in each package with some level of specificity.

Measure could be a self-report like survey (mature) and observation and assay (less mature).

This would need to be tweaked in line with certain design principles we develop.

Achim was thinking of a layered model. At the lowest level are UML data primitives. The next level is foundational objects. Then there are things like concepts, variables, etc., at the third level. These are building blocks that could be used as packages. This could be the middle of the table on the slide. The right hand could change according to data type.

Are these good terms? It is difficult to agree on terms that all agree on. Would this model be useful in creating packages? It provides a theoretical framework so it enhances communications and helps people see the whole. The language is sufficiently neutral so that people can engage with it.

**Study Inception – Data Management Plans**

This small group developed some XML for their content, created an XSLT spreadsheet, and then showed the output of what they are calling Study Inception. They quickly realized that they needed the sampling and methodology information that has already been created to be imported into Drupal for purposes of their content area since study inception overlaps with these areas. They also recommended that DDI describe field work -- we need a process model for this.

A Web application, possibly made available via Drupal, would be useful so that XML can be generated.

**Documentation**

There is no way to validate a view right now. We are doing documentation in Drupal and have tested if we can export it as Docbook and we can. Syntax binding can act on the XML and inject it into the schema. But we have never been clear about the exact flow of how documentation gets into the schemas.

How do we get documentation out of Drupal and into Enterprise Architect? Documentation flows from Drupal into Docbook in a form out of which we can create publishable documentation. How do we get documentation into syntax annotation? We could use this Web form and add the capacity to carry Docbook. With unique names we can pull documentation together.

**Consolidating Platforms**

Currently, DDI uses several platforms for different purposes. We need to consolidate these platforms, which will involve moving published versions from Sourceforge to Stash. Using the suite of Atlassian products (Jira, Confluence, Stash) will help us consolidate. The table below shows the multiplicity of platforms currently used:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Drupal | EA | Assembla | Mantis | Website | Sourceforge | Google |
| Content capture; XMI generation; Transparency/ communications; documentation; basic modeling | Model validation; content creation RDF, XML | Subversion site; archiving XMI from Drupal; historisation | Issue tracker; future issues; hist. issues | Published docs; comms to externals; CVs | Previous published specsXXXX | Word docs, etc. |
|  |  |  |  |  |  |  |
|  |  | Stash/Bitbucket | Jira | Links to stash | XXXX | Confluence |
| Files | Files |  | Future issues, bugs |  |  |  |

The group looked across all the platforms. It was decided to mirror Drupal and EA output to Git using Stash, which is an Atlassian product, to use Jira for issue tracking, and to move sourceforge published specifications into Stash. There was a discussion of whether to integrate the DDI Website into Confluence. We would use Confluence to manage Word docs.

The systems do require a commitment in terms of administration, maintenance, and implementation. We need people to configure, populate, and update and the systems. Do we need to pay someone to do this? Setting it up initially might require a lot of work.

Using Confluence as the external facing Web site was discussed, but it is not as sophisticated as Drupal, which is the current technology used for the DDI Alliance site.

For internal documents, we need spaces for groups (publicly available) and a commenting facility either with access control or for the public.

We need to figure out hosting. The open source route means that all content has to be publicly accessible.

We will need to make a decision about hosting and determine who does the updating and who does the configuring. We can send the table and relevant questions to the technical committee and to the individuals who have volunteered to contribute on this. Also we should now download the trial version and set up a trial instance.

**Study Inception**

The Study Inception-Data Management Planning group reported back to the larger group. They added many new objects as they went through the UK’s Data Management Plan template in detail. What came out of the exercise was XML for a Data Management Plan and more general information around study inception. The next step is to compare this against the NSF and NIH data management plan requirements. This could be two views on the same content.

**Archive and Library**

This group reviewed the Archive User Story and added existing elements from 3.2 as well as elements that are missing. Much of the missing content relates to the process model that needs to be developed.