Sprint Report

DDI Moving Forward | Copenhagen Sprint | 23–27 November 2015

Participants: Arofan Gregory, Jannik Jensen, Joachim Wackerow, Jon Johnson, Marcel Hebing (sprint master), Oliver Hopt, Olof Olsson, and Wendy Thomas

Executive summary

The group for this sprint consists of two sub-groups: one group (the developers) working on the implementation of our production workflow and a second group (the designers) working on related design decisions. On the practical side, the most important output of the sprint is that we now have a working production workflow that automatically generates our products (XSD and RDFS/OWL) as well as the documentation (HTML and PDF) from Drupal and related sources. On the design side, we were able to answer fundamental questions regarding the overall design of the production framework as well as more detailed questions and issues. It is worth mentioning, that this sprint was quite small regarding the number of participants (2.5 developers, 1 sprint master and 4 others). We are proud that we got a substantial amount of work done despite the small size of the team.

Introduction

This report consists of five parts. We first discuss the four priorities of the sprint. After that we take a closer look at the outputs from (1) the developers group and (2) the design group. Regarding the organization of the sprint, we document the schedule. Finally, we take a look at more detailed issues, which we gathered in a question-and-answer format—some of

these issues are considered to be done, while others are reported back to the respective groups for further consideration.

Priorities

In Dagstuhl, we prepared a list of four priorities for the Copenhagen sprint. The following overview presents the four priorities in descending order and explains what we achieved regarding each of them:

Priority 1: Completely automate the production workflow. The production system is to be driven by a build server, Bamboo, to perform a series of functions in sequence, producing all of the work products and documentation. Getting the automated build working is critical for the further development of DDI 4. In this sprint, we implemented a fully automated build process based on Ant. The use of Bamboo for our build server is still challenging.

Priority 2: Design work regarding the production workflow. The implementation of the production system was accompanied by the work on various design decisions. This included a definition of functional views, the review of the RDF feedback from the Dagstuhl sprint, a review of the XMI and UML we are using, the definition of binding rules for both XSD and RDFS/OWL, and many other topics. The full list is presented below in the design outcomes section.

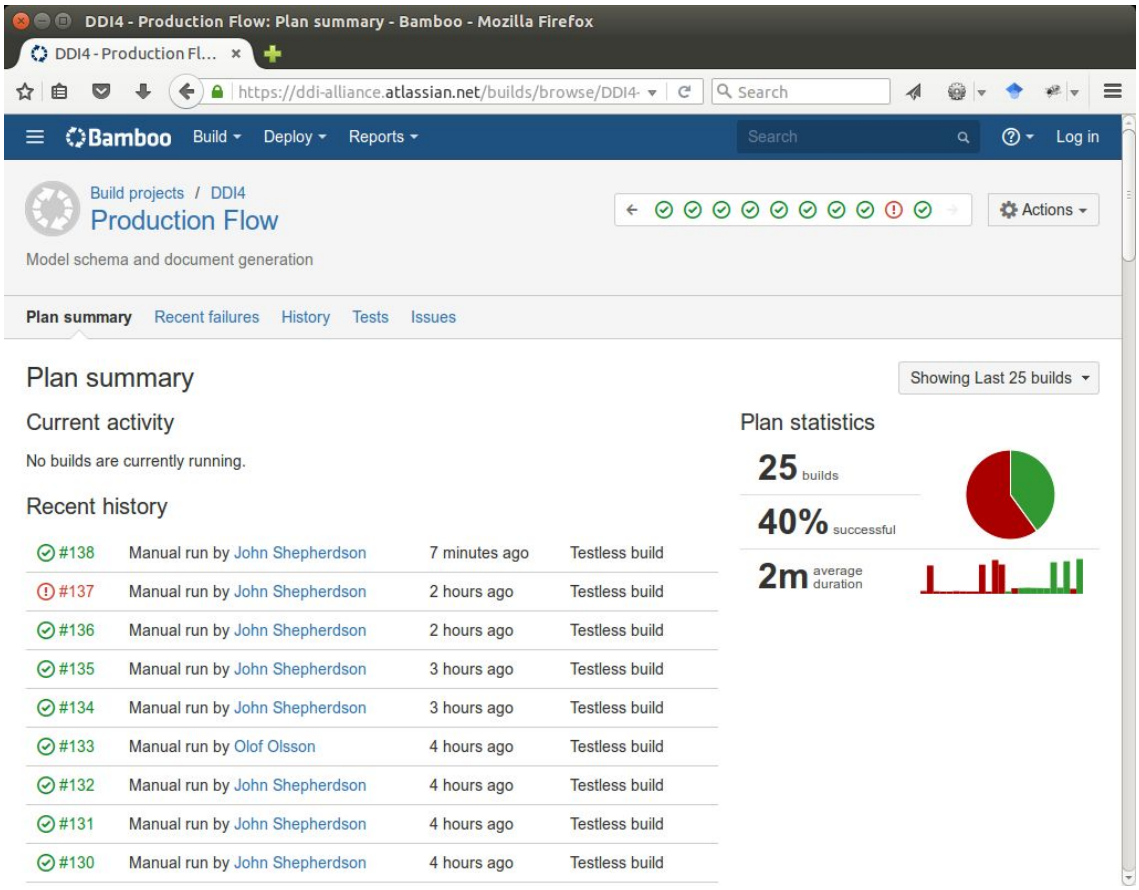
Priority 3: Improve the individual tools (Drupal, bindings, etc.). We made basic adjustments to the build tools like splitting the binding process according to the design of PIM, PSM, and schemas.

Priority 4: Clean up (content and code). To get the binding scripts up to work, some basic cleanup of the Drupal content was necessary. This discussion resulted in a document defining issues and solutions for a clean model in Drupal.

Developer outcomes

The most important goal of the sprint was successfully met—below can be seen the results of a production run in Bamboo, showing the creation of work outputs including all XSD schemas, RDFS/OWL specifications, image files, and other deliverables.

Figure 1. Success!



To achieve this, much time was spent working on the configuration and operation of the build server. In addition, the generation scripts for both XSD schemas and RDFS/OWL specifications were adjusted and corrected. The move to the introduction of two-part production flows (PIM-to-PSM, PSM-to-XSD/RDF) was implemented. This work also involved some clean-up of the model in Drupal. Some minor issues remain regarding Bamboo, resulting from issues with the software—these have been reported to Altassian.

The significance of having a functioning build with the creation of XSD schemas and RDFS/OWL specifications is particularly significant, as it allows examination and

assessment of the products, with potential implications for how the model might need to be adjusted. This has been much demanded by both the Modelling Team and the Technical Committee in order to support their work.

Related links:

- ➔ Production framework on Bitbucket
- ➔ Build server on Bamboo

Design outcomes

Topic 1: Functional views. In order to create a binding of a Functional View we discussed the three questions: (1) How will the Functional View be constructed (technically)? (2) What "Functional View Level" documentation is needed and where does this reside? (3) What set of standard content classes should be available in all Functional Views or identifiable sets of Functional Views?

Output document: Functional view

Topic 2: Revise Michel Dumontier's feedback on RDF. We followed up on Michel Dumontier's feedback in the 2015 Dagstuhl sprint. Based on a first list of issues composed by Achim, the group discussion goes into more detail.

Output document: RDF notes

Topic 3: UML and XMI. This topic combines three issues: (1) We want to define an XMI flavor for DDI, which confirms to the standards UML and its representation XMI, is robust, and can be imported into major UML tools. (2) The definition of a subset of UML 2, which we use for the definition of the PIM. (3) Do we want to publish our XMI files as an additional work product? A business case for this was drafted and sent to the AG for consideration.

Output documents:

1. XMI flavor for DDI
2. UML constructs
3. XMI as a work product + long version

Topic 4: Local use of the build process. The intention for this discussion was to ensure that interested people should be able to run the full build process locally for testing purposes. In particular, it should be possible to use any XMI file as an input. Due to the flexible design of the build process, this is now possible.

Topic 5: Update the design of the production workflow. To update the production workflow design has become a continuous task. Furthermore, we prepared a framework (product matrix) and checklist to define the product of DDI 4.

Figure 2. Concept for the production workflow.

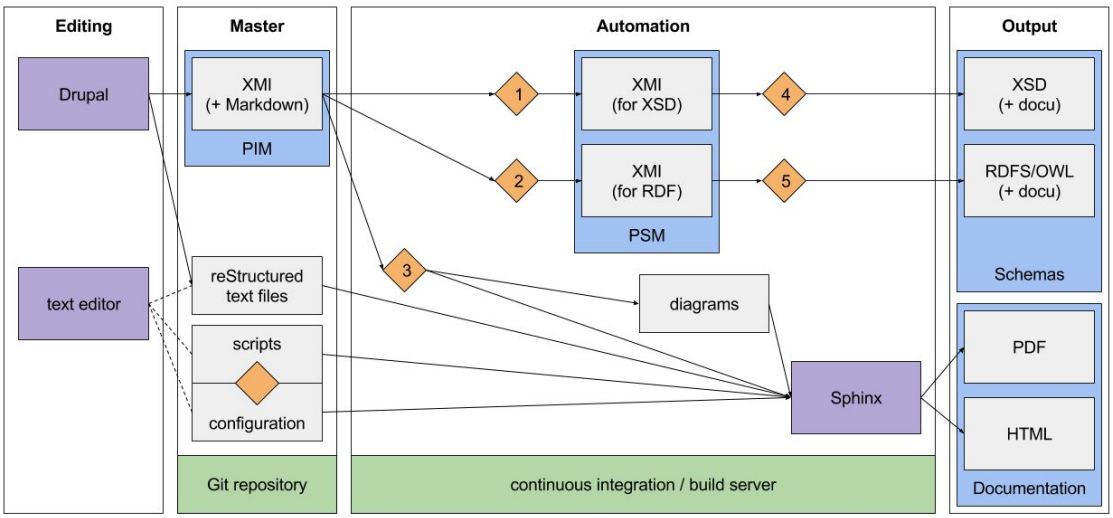
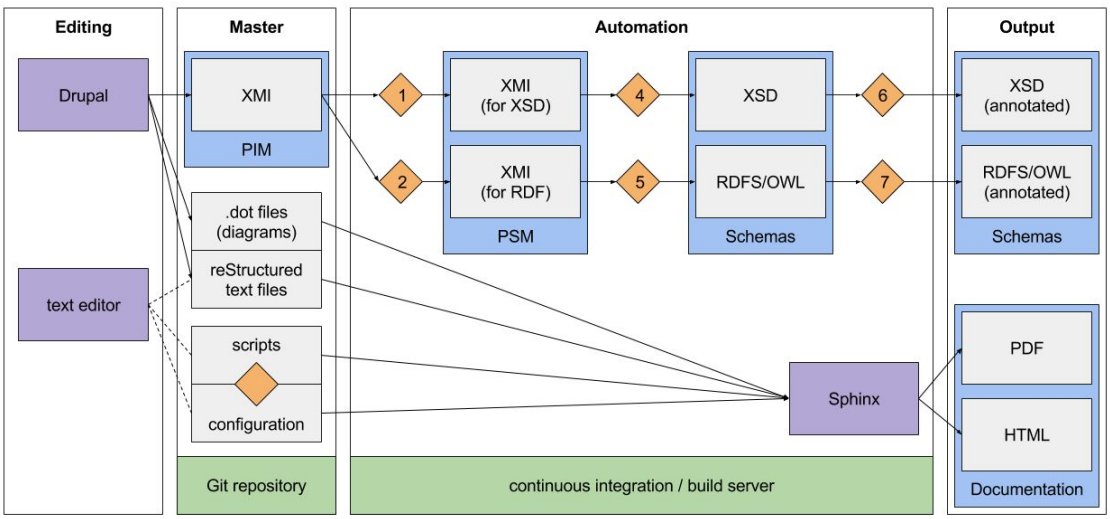


Figure 3. Status quo of the production workflow.



Output documents:

1. Production workflow - concept
2. Production workflow - status quo
3. Product definition

Topic 6: Definition of binding rules. After creating a framework for binding rules, we defined the rules for XML and RDF. In addition, we talked about RDF-XML-namespaces to take an important step towards round-tripping.

Output documents:

1. XML Binding Rules
2. RDF Binding Rules
3. RDF-XML namespaces

Topic 7: Outline for the documentation of design decisions. How do we document the design decisions applied in DDI 4 and the production framework? The output of this discussion is a draft / framework for a more comprehensive documentation.

Output document: Design Decision Framework

Topic 8: Design of the build server. Complementing the implementation of the production workflow and setting up the build server, we discussed requirements for the build server and a definition for the repository structure.

Output documents:

1. Requirements
2. Repository structure

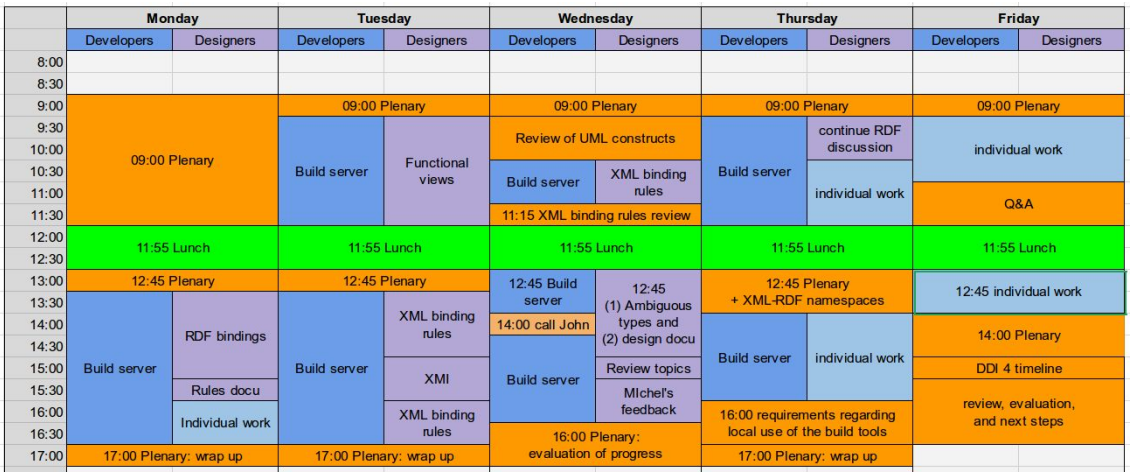
Topic 9: Clean up of the model. In order to implement the binding scripts (both XSD and RDFS/OWL), we required certain decisions to clean up the model.

Output document: Cleanup decisions

Schedule

Work was performed in plenary and in breakout groups, as can be seen in the following schedule. The facilities at Statistics Denmark were fantastic!

Figure 4. Work Schedule.



Questions and answers

This section provides an overview of more detailed questions we identified and worked on. The full list of questions with more detailed comments and answers is available in the following document: Questions and answers

Done

- ➔ What do we like to archive with the RDF? (Jon)
- ➔ How to document the transformation rules, the design decisions behind? And what are the outputs?
- ➔ What is the scope this week regarding the build?
- ➔ What is the first use case (functional view) to implement?
- ➔ How do we drop attributes in functional views?
- ➔ What does the functional view look like in Drupal (diagrams, usage information, etc.)?

- ➔ Are substitution groups necessary for dynamic text (where order is important) in the XML and RDF binding?

Open questions for after the sprint

- ➔ Do we stay with XSLT for the XSD generation?
- ➔ How to do cleanup of unused stuff in lion?
- ➔ How can the test suite be integrated in the build process?
- ➔ As DataCapture is the most mature, it could be a good test case (+ related stuff). How can this be integrated into the build process?
- ➔ Should the logs and outputs from the build server be publicly available?
- ➔ Who can do the following tasks?
- ➔ How do we get the embedded documentation into the XSD / RDF?
- ➔ How to make the data type definitions in XMI (Currently using class definitions…)? (Achim)
- ➔ How do we handle external references in the XML (and RDF?) binding?
- ➔ Completion of export of class level documentation from Drupal in reStructuredText
- ➔ Formatting of PDF to publication quality standard
- ➔ Specification and generation of convenience RDFS/OWL “schemas”
- ➔ Issues include using OWL Ontology, named graphs
- ➔ Review of RDFS/OWL schema definition