# Minneapolis Sprint, 25-30 May 2014

# Hosted by the Minnesota Population Center, University of Minnesota

**PARTICIPANTS**

Anita Rocha, University of Washington

Arofan Gregory, Open Data Foundation

Barry Radler, MIDUS - University of Wisconsin-Madison

Dan Smith, Colectica

Daniel Gillman, U.S. Bureau of Labor Statistics

Flavio Rizzolo, Statistics Canada

Jay Greenfield, Booz Allan Hamilton (BAH)

Jeremy Iverson, Colectica

Joachim Wackerow, GESIS - Leibniz Institute for the Social Sciences

Johan Fihn, Swedish National Data Archive (SND)

Johanna Vompras, University Library Bielefeld

Larry Hoyle, University of Kansas

Marcel Hebing, German Socio-Economic Panel Study (SOEP), German Institute for Econmic Research

Michelle Edwards, CEISER

Oliver Hopt, GESIS - Leibniz Institute for the Social Sciences

Olof Olsson, Swedish National Data Archive (SND)

Simon Lloyd, Australian Bureau of Statistics

Steve McEachern, Australian Data Archive at ANU

Wendy Thomas, Minnesota Population Center (MPC)

Anders Swendsrud, Minnesota Population Center (MPC)

**METHODOLOGY SUMMARY**

Reviewing the model created in Dagstuhl, we defined ‘x’ in the model as the outcome of a process. Using case studies for weighting, coding, sampling, harmonization, and confidentialization, we examined whether we could document those methodologies with the general model. We determined that ‘design’ still needs to be defined more explicitly. While we recognize that ‘process’ represents the process model, we modified the model to provide the option of expanding the process for any number of process steps. ‘Design’ will now be an extension point allowing users to extend customized designs. We recognized the need for adding ‘other material’ and ‘citation’ to the model. In the future, we recognize the possibility that the extension point will include models for sampling, weighting, and sub-models previously developed by the SDI group such as those for para-data and questionnaire instrument.

**PRODUCTION TEAM SUMMARY**

The Production Framework Team worked this week with the Modeling Team to address several important issues. The Drupal Development team worked separately on resolving open issues, and will report separately. The areas of significant progress were first, the clarification and improvement of the transformations for producing syntax bindings, and second, the design of the XML binding itself.

Improvements to the process of creating bindings from the model included those to the documentation and specification of rules in a human-readable form, and the introduction of a platform-specific model generated from the conceptual model. This simplifies the syntax binding, and makes the entire process more transparent. The design of the XML binding and how the use of XML schema were examined in light of emerging developer requirements. The resulting proposed solution addresses several compromises in the earlier design. These allow for a "have your cake and eat it too" scenario, in which different competing requirements can be met simultaneously. The XML schemas will support implementing single views, implementation of the full library, and even the creation of interoperable custom views, easily created by users.

Remaining major areas of discussion include the completion of the discussion around versioning, in light of progress made in the area of the binding process and the XML design.

**DATA DESCRIPTION SUMMARY**

The original goal for the Minneapolis Sprint: To be able to describe a CSV file by its structure.

We came up with a generic model (a picture of which is included in the Meeting minutes) dividing the basic data description into logical record and physical structure. The emphasis of the meeting became on refining the logical structure to incorporate the datum based structure that had been in development since the 2014 Dagstuhl meeting. This datum based model was refined on Day One and Two.

We then discussed whether the generic model was appropriate for 1) a CSV file and 2) an aggregate data (n-cube). We started with datum, but we took advantage of the work already done to describe datum. We came up with original definitions for record, record set, and data store, but those were revised on Friday morning after further discussion. By Friday morning, we created a new object called datum structure which is made up of the key, event, value, variable, and universe. This structure will be incorporated into Lion as soon as possible.

Definitions for the following were developed (currently subject to review):

A **record** is an ordered set of tuples, all of the same key and universe.

A **record set** is a set of records that all share the same universe.

A **cell** in a cube is a set of tuples all with the same key, event, and universe.

Questions or observations for future discussion included:

1. Does our definition of a record fit all possible use cases? For example, does it apply when one record is equal to an array of n-cubes?
2. We need a clearer definition of event.
3. Can we fit para-data in our general model?

The group will continue working after the sprint on the aggregation of the datum model into the larger aggregates such as unit record files and dimensional data.

**DRUPAL SUMMARY**

New features

* Graph improvements
* Mappings between objects in model and 3.2/GSIM more visible and added to Docbook documentation
* Added inherited properties and relations to class pages
* Added a new content type to Drupal which is used to hold the content of the front page of Docbook documentation
* Various Docbook issues
* Various bug fixes

Suggestion for documentation

* Use Sphinx as a complement to Docbook to produce better looking and easier browsable documentation

**CITATION SUMMARY**

 The Enhanced Citation model was presented in several venues and comments were considered in revising the model. This included:

* renaming a number of objects
* using the revised model for OrderRelations
* adding a hasConcept relationship to CustomItem
* adding a key property to CustomItem
* adding a key and a value property to  CustomValue
* setting CustomInstance and CustomValue to inherit from Identifiable

The models for custom metadata and controlled vocabularies have been entered into the Drupal site (lion) as the CustomMetadata  package. We discussed adding a relationship to CustomInstance to Annotation but that has not yet been done.

We also discussed which parts of the openEHR blood pressure model might be described as custom metadata and which parts might be collected as data. An intended protocol description might be made using custom metadata with keys of CuffSize, Location, Method, MeanArterialPressureFormula etc. as keys in an openEHR related CustomStructure. At the same time the data description group discussed how some of these same elements in other situations (e.g. where the protocol might vary for each patient) might be collected in event tagged DatumStructures.

We also discussed developing an example of documenting a Question with custom metadata structured in relation to the U.S. Office of Management and Budget requirements for collecting information such as

* OMBNecessity -  Why is this data collection necessary and how will the information be used?
* OMBFrequency - How often should data be collected?
* OMBBurdenHours - What is included in the calculation of burden hours?

**DATA CAPTURE SUMMARY**

The ostensible goal of the Data Capture group was to extend our previous work on Simple data captures or instruments to Complex ones. The Simple-Complex dichotomy has long been considered a false one, in that there are no clear distinctions or characteristics that define a simple instrument from a complex one. Creating an information model for Data Capture has only further clarified this, in that the model developed by our team the past year and half is not perfect, but is relatively comprehensive, parsimonious and (so far) robust enough to document many different data captures of differing complexity. We have been conscious of defining data capture broadly to include non-survey scenarios and processes – blood pressure data capture continues to be a useful exemplar and use case to model for our group and others.

DataCapture already had applied use cases of different complexity to the model during the Dagstuhl sprint. For the Minneapolis sprint, the primary goals were to compare the DDI 4 capture model to relevant DDI 3.2 objects and reconcile any differences. The work begun here was documented in a cross-walk and could prove a useful template to other groups in ensuring the backwards compatibility of DDI 4 with 3.2, as well as identifying any gaps or oversights in DDI 4 models. This exercise resulted in (1) renaming of the Capture object to Measurement, and revising its properties to clarifying its broad non-survey nature; (2) specifying the properties and definitions of (nearly) all objects in the model; and (3) tweaking the cardinalities so that most objects are not required objects. Discussion was also initiated on the relationship between Capture and Methodology – there seems a relatively clear “fit” between the two at the Conceptual (Design) and Sequence (Process) levels.

Outstanding issues are:

* Clarifying the nature relationship to Data and Process models; further work awaits the refinement of these models
	+ The idea was floated during soap box that a small number of “standard” use cases should be created that are applied in each of the DDI 4 information models. This will ensure that ach group is not being developed in isolation of each other.
* Clarification of ResponseDomain and how this object (Representation model?) will be used to define data type
* Nothing much was resolved about the role that DDI 3.2 objects QuestionGrid and QuestionBlock will play, if any, in DDI 4.

**SIMPLE CODEBOOK SUMMARY**

Due to resource constraints, including participation of team members in other Sprint activities, the Simple Codebook group did not have a formal meeting during the sprint.

A small group consisting of Steve McEachern, Anita Rocha and Michelle Edwards did however get the opportunity to review the Methodology section of the DDI Codebook (2.5) model and compare the content with the current activities of the Methodology group. The group identified several key areas that will require content to be developed by the Methodology group to enable support for DDI-C, and documented these in the Methodology team minutes for the afternoon of May 27th ([https://ddi-alliance.atlassian.net/wiki/display/DDI4/Methodology+Team+Minutes](https://ddi-alliance.atlassian.net/wiki/display/DDI4/Methodology%2BTeam%2BMinutes)). There was also some discussion of the need to update (or potentially deprecate) some fields inside the DDI-C, given changes in terminology that have occurred since the original release of DDI-C in 2001.

**KANBAN**

DONE

* Methodology/Process
	+ Map sampling model against Methodology Model
	+ Assess current Methodology Model
	+ Develop questions for MPC team for Thursday
	+ Test generic methodological design model with MPC team
		- Confidentialization
		- Harmonization
* Data Description
	+ What left to do with data description, what needs are there?
	+ Logical Model
	+ Basic build block
* Drupal
	+ Duplicate values in xmi
	+ Revision log
		- Expose on every page
		- Required use of revision log message
	+ DocBook
		- Output sections sortable
		- Weight field added
	+ Add build date in DocBook export
	+ Warnings for duplicate names
	+ Add mappings to GSIM/DDI 3.2 to DocBook
	+ Show parent properties and relationships inline
* Production
	+ Decisions on style of XML binding (prototype)
	+ PIM-PSM intermediate model for binding document
	+ Document build server discussion
	+ “What is a view” documentation
* Data Capture
	+ Update implemented instrument
	+ Update conceptual instrument
	+ Update question

IN PROCESS

* Methodology/Process
	+ Map out generic design model for methodology
	+ Develop objects for description of design:
		- Weighting
		- Coding
		- Harmonization
		- Confidentialization
* Data Description
	+ Aggregation into data structures
		- Use CSV case, logical structure, and physical structure
	+ Model unified data structure – key-value tuples
	+ Describe a simple type (output for next release) (tidy it up)
	+ Model “unit” for microdata and dimensions
* Drupal
	+ Sphinx proof of concept
	+ Render external objects as full objects
	+ Other DocBook bugs
	+ Add graph by class
* Production
	+ Production group to discuss versioning
	+ Document of more specific views in XML binding

TO DO

* Methdology/Process:
	+ Review input, output, binding
* Data Description
	+ Add Data Structure objects to Drupal
	+ Extend to complex structure
	+ Harmonize Dagstuhl and MPLS objects
	+ Representation vs. Data type
* Data Capture
	+ Add question collection
	+ Address block and grid