DDI Moving Forward, Sprint #1
Monday, October 28, 2013

Introduction

The week began with a series of presentations orienting sprint participants to the DDI Moving Forward process, the sprint process itself, and the goals for the week.

The goal of DDI Moving Forward is to build the next-generation DDI specification based on a data model. The objectives are to:

- Improve DDI usability
- Expand DDI coverage
- Express the model in different technical formats
- Repackage the specification
- Enhance the documentation

There is a lot to accomplish in a short time so the sprints are designed to push the work forward quickly. Five sprints are envisioned:

- Dagstuhl -- October 2013
- EDDI – December 2013 (Paris)
- NADDI – April 2014 (Vancouver)
- IASSIST – June 2014 (Toronto)
- Dagstuhl – October 2014

There will also be virtual work taking place between sprints with the goal of a first version of the DDI model-based specification available March 31, 2015.

The scope of the Dagstuhl sprint is to model four function areas:

- Simple data file description
- Foundational metadata (variables, concepts, categories, etc.)
- Common codebook (commonly used profiles)
- Simple instrument

The Sprint process

The sprint terminology comes out of agile software development. Sprints are self-organizing with frequent plenaries and flexible assignments. It is essential for the group to be productive and to make good use of the time together.

Deliverables

An ambitious set of deliverables was outlined:

- Completed model templates for the four identified functional areas
- First draft of common model documentation based on completed templates
- Draft design and specification of production process
- Draft design principles for DDI work products
- Documentation of DDI Alliance workflows
- Draft XML syntax binding specification
- Draft RDF syntax binding specification
- Draft business rules specification for model transformations
- Documentation of modeling style (includes UML profile)

The group working on content in the four functional areas can draw on existing the Generic Statistical Information Model (GSIM) Version 1 and earlier versions of DDI.

It is envisioned that the new DDI will have a core plus extensions and that small functional chunks of the model will be released on a staggered schedule. DDI will not be a monolithic whole any longer and there will be no "big bang" releases.

What comes out of the sprint should be an implementation model. The use cases done in the areas of archives, questionnaires, and longitudinal studies feed into this modeling work. We also want to think beyond survey data to new data types and to challenge our preconceptions.

Technical work

We will be developing a UML Class Model, which can then be transformed into XML and OWL. Thus we need generic rules and specific binding rules (configuration). We can also think ahead to prototype representations of the model in Java/ C#/DBMS/DDLUtils, although this will not be a focus.

Disciplined use of UML is an answer to the potential complexity of the modeling task: modelers can use a specific set of components. UML and XMI are complex and have several variants. For the modeling work we will use Enterprise Architect (desktop version) and will produce UML model definitions and visual representations.

Documentation

Good documentation is essential and having the model be self-documenting is the goal. During the week we will produce one document per object outside the UML tool and this content will feed into creation of the model.

We will ultimately need platform-independent and platform-dependent versions of documentation. We would like to use an XML document format to enable efficient reuse and are looking at DITA or DocBook for this purpose.

The content groups started out using Word templates but during the week a Drupal form was developed to provide structured export.

Open issues

- Export of XML needs to be carefully checked
- Explore combination of visual model and textual definition (outside of Enterprise Architect)
- Modularity in exporting from the model needs to be explored

OWL is more expressive than UML. We will need to do some handcrafting of specific binding rules. There may be multipart transformations to OWL -- autogenerated parts and handcrafted relationships to other vocabularies.

There is a lot of work to do. We will need to produce hundreds of object-level documents as input to the modeling work.

Terminology

A clarification of terminology was provided:

- Conceptual model – High-level business level (GSIM); can't be implemented directly
- Reference model – Synonymous with conceptual model
- Implementation model – Platform-independent but implementable
- Application model – Platform-dependent
- Information model – Metamodel and probably the term we want to use for DDI

Initial strategy for organization

The group broke up into two large groups to start the work with the knowledge that smaller groups would form during the week. The two groups were:

1. Content modeling (G. Duffes, D. Gillman, L. Hoyle, J. Johnson, M. Karjalainen, J. Linnerud, S. McEachern, B. Radler, O. Risnes, W. Thomas, M. Vardigan, W. Zenk-Moeltgen)
2. Technical -- Bindings and technical production flow and modeling style (T. Bosch, R. Cyganiak, J. Fihn, A. Gregory, O. Hopt, S. Kuan, B. Mathiak, O. Olsson, J. Wackerow)

Content Group

GSIM

The group started its discussions by asking whether our effort should use GSIM for its foundational metadata. The group thought this was a good idea since we need to align with GSIM.

A variable in GSIM is a description of a variable in a dataset (instance), but we may want to start at a higher level.

Concepts and roles

The group started by modeling three roles of concept guided by Dan Gillman, who was involved in the GSIM effort. The roles are: Category, characteristic, and universe.

Universe: This is equivalent to the GSIM population. These are the objects one is studying. It is akin to units of analysis.

Characteristic: This is equivalent to a variable. A variable provides characteristics of a unit type like sex, marital status, etc. In GSIM there is an Instance Variable and a Represented Variable.

Category: How the characteristic is measured – e.g., Male and Female. Categories can be described or enumerated.

Sanity checking with other data types

Because we want to move beyond documenting survey data with the model-based DDI, the group considered other data types and whether the use of concept with its three roles would be sufficient to represent other types of data.

Qualitative data

It was decided that qualitative data fit in this scheme also. Two types were discussed: (1) unstructured text from an open-ended question – "Was there a certain event that made you aware of the recession?" and (2) a collection of photographs.

Answers to the question are essentially long text strings and the universe is people. Conceptually there is a reaction to the event – a personal account of reaction to an event (representation of the concept).

We can distinguish three types of qualitative approaches: characterizing some body of data like photos, defining segments on the qualitative objects, and applying quantitative procedures to a body of unstructured text (text mining). In qualitative methodology, researchers often work "backwards" and may apply concepts later. Qualitative researchers often let the objects steer them in their analysis.

A description of photo might be qualitative and this fits with our model of concept and its three roles. A researcher might move from qualitative to quantitative because in essence he or she has classified the text. The researcher doesn't know the concepts when starting out but does when he or she records the result. Metadata around photos can be data. Each analysis process can be a process of data production with variables coming later.

Experimental design data type

The group considered experiments from economic sciences – specific groups in a laboratory playing games with the goal of determining what the best strategy is, or game theory. People come from a universe, and we can categorize people by different conditions with different outcomes for different groups. The experiment is a tool to test behaviors and to answer a research question. The universe could also be a set of behaviors with reaction times, triggers, etc., as characteristics (behavior can be a universe or characteristic).

Characteristic and category are things you record for each of the groups to indicate behavior. A machine or an observer can capture data also.

Timebound events data type

One might measure the height of a plant every day for 10 days. The variable itself is not getting redefined. We might also capture marital status continuously. There will be date-time stamps and status changes. All concepts are implicitly bound in time and geography. There may be relationships between characteristics.

Documenting process

A question arose about whether the group should be documenting process. This is important but is separated in GSIM for a good reason. We decided we would not consider this now.

Concepts, universes, designations, categories

Universes are the things that we measure: actor, utility, and event/process/activity can all be measured. Representation happens when you care about making a datum and writing it down. A code is a particular representation of a concept (known as a designation).

A designation is a representation of a concept by a sign (string, bitmap, pictogram, etc.) that denotes it.

A question was raised about whether we need to add Category Set to our model. Categories exist in a set and the set itself is a concept. Do categories exist

independently of the set? Most people say no but it's complicated. A category can exist independent of the set and can be reused. A conceptual domain is an enumerated set of categories or a description of what they are. Levels involve classifications. The Neuchatel model for classifications is being updated to be part of GSIM.

What is data?

A datum is a designation – a unit associated back to a category in a category set. Data come into being as the result of a determination. A characteristic is that which is capable of being determined and a category is the answer to that determination or a determinant.

DDI "administrative" metadata

The group considered whether we need to retain the DDI approach of Versionable, Maintainable, and Identifiable. It was pointed out that in other systems, versioning is handled by actors outside the model. We should not take as a given that we need to keep these administrative metadata items.

Adding new objects

The group decided that Designation and Sign were needed as foundational objects.

Technical Group

- Need to decide on types of modeling for the UML Model
- First priority: List of requirements of what the production system should be
    1. System should support improved deliverables
    2. System should be reasonably flexible
    3. System should be robust and as simple as possible
    4. System must be sustainable (with changing people)
    5. DDI Alliance must be developable + maintainable with existing resources
        - Easily maintained – by different people
    6. Must capture intelligence of biz/content teams as first step
        - Biz people must be happy (tooling must be easy to use)
        - Must be known process
    7. Documentation must be closely linked to model but support parallel work among several teams
    8. Must support modular output on different platforms

(module = package in UML; namespace in XML, etc.)

9. System should support re-usable rich text (HTML) for all production outputs (must be identifiable at a granular level and be machine-process-able)
    - Documenters must have a usable interface (but product should be machine-processible)
    - Initial documentation must be auto-generated from UML to the extent possible
    - Some suggestions for capture (nice to have on-line or off-line)
        - Informal work post-processing (DocBook)
        - Structured Wiki
        - Front-based tool (XML or Relational-based)
        - Google docs?
        - Develop custom tool
        - Would generating an XMI output from the capture tool be useful?
        - A tool that can accept XML and output something in different format (e.g., XML)
10. System must maximize automation
11. Automated management of collection of products
12. Transparency at each step of production process must be supported
13. System must support versioning of modules/release + packaging process

- With this approach, how do we determine compliance to DDI?
    1. Will have 2 canonical outputs for compliance (RDF, XML) – take "modules" and turn into XSD for example

Action item(s):

- Ask content people in plenary whether the capture process needs to be available off-line, plus other capture-tool preferences
- Ask documentation people in plenary what their requirements are (offline/online? XML editor OK? What formatting tool for creating publishable outputs?)
- To-Do in Committee: Can content people and documentation people use the same editing tool?
- Review some of the options proposed for data capture -- e.g., DocBook, Wiki, etc.
    - What Wiki platforms are capable of structured information capture?

    - What word-template solutions exist for DITA? What for DocBook? Are these useful tools?
    - Are there forms-based solutions for each standard, and what do these look like? HTML forms? XForms? Proprietary Microsoft products? What is the effort to develop the needed structure-capable rich-text editor?
- What are the requirements for technical material such as RDF-specific examples and XML-specific examples? Are these areas of documentation edited by the same people who do other documentation, or by people who specialize in these technologies?
- Parking-lot Issue: Do we need unique names (e.g., for variables?)
- To-Do in Committee: Need to find out more about requirements for bindings.
- What subset of DocBook would be useful for our documentation needs? What subset of DITA would be useful for our documentation needs? What are the structures in our documentation (ask documentation people in TC)? Do we need rich-text capability?