How to review the Q1 Development Draft

General Instructions:

The intent of this review is to check in with DDI users on our progress to date. DDI 4 involves a new model driven perspective, a more production focused development structure (i.e. bindings and documentation are produced from the model), a focused set of products, and greater ease of use. As such we need feed-back on what we are doing (the model and documentation) and how we are doing it (the approach).

This first release is limited and not intended to be used as a basis for implementation. It is a Development Draft of a core portion of our work to date. Therefore it does not cover more than a subset of classes that are used in many applications of DDI: how multiple languages, dates, and commands are captured; how classes are identified and annotated; how agents (people, organizations, and machines) are described; how basic structures like Concepts, Universes, and Representations build toward describing specific variables; and how to create structured collections or groups of similar classes. Two Functional Views using only classes found in the Q1 release show what a DDI user would see if selecting a Functional View to meet a specific need. For example: Agent View is intended to be used to manage a set of organizations, individual, machines, and their relationships to each other. Discovery View manages a set of basic bibliographic information.

We have not published the bindings (XML Schema and RDF/OWL Vocabularies) in this release. These will be released at a later date and we will ask for comment on them at that point.

Below are some specific issues you may wish to comment on using the comment form provided. We are interested in all of your thought on whatever aspects of DDI 4 you wish comment on.

Reviewing the approach:

- Does it have reusable components to the extent needed?
- We have some abstractions of things, like collection and other patterns? Is this the way to go?
- Is there a balance between real life and a clean model in a technical sense?
- Are the classes modeling something in real life? Is it clear what a class is and what it relates to in the real world?
- Reaction to the cascade model of describing variables

Reviewing the model:

Library Packages (Classes):

- Are the set of properties complete? Are thy under-specified, too simple? Over-specified, too complex? Can you provide a sense of the percent of coverage we are providing in terms of what is needed to describe a class?
- Do properties and relationships look correct?
- Are there too many levels in terms of relationships?

- Are the classes modeling something in real life?
- Is there a balance between real life and a clean model in a technical sense?
- Does it have reusable components to the extent needed?
- Is the class level documentation sufficient? Do the classes relate to the domains well? Do we understand how they relate to other domains?

Functional Views:

Functional Views provide focused sets of classes to address a common use case or viewpoint, for example, a question bank, a simple codebook, a simple instrument, etc. Note also that we plan to release a Functional View containing the whole library.Is this a workable approach? Is it sufficiently documented?

- Are the right classes included in each Functional View given its described coverage?
- In the future when we publish other views we intend publish in this format but with use cases, narratives, how to navigate the contents, etc. We want to provide information to help people think about their cases and decide which Functional View applies and how to use it. What would be useful to have?
- Although the current Functional Views are very limited, do you see yourself using them? Why or why not? What could be done to make Functional Views better?
- If you don't plan on using this (a DDI structure to manage the metadata found in the current Functional Views), what would you be using? Let's us know what we should be looking at in terms of what we cover.
- In DDI-C we had top level elements (codeBook or DDIInstance) which contained a consistent set of information. XML and RDF handle this in different ways. This topic is still under discussion within the Moving Forward Project. Is there a set of information that is needed for all or most instances of a Functional View?

Reviewing the documentation:

- The documentation files are a work in progress. What information do you need that is not there? What is “too much information”?
- How could we make it easier to use? To understand?