DDI4 Goals:[currently contains contents of two base souces 20171016]]

**Why a new version of DDI**

DDI has continuously evolved to meet the needs of its user community. The development of this latest iteration was begun in earnest in October 2013 at Dagstuhl and has continued through a series of sprints and online meetings.

**Goals**

To produce a model-based specification that can be more easily managed, extended and be expressed in different representations

To be more responsive to changing requirements from the user communities

To more easily align the specification to other current and emerging standards

**Why we're doing it**

● In reaction to the new requirements of both data and the processes around data collection and acquisition, the standard has got more complex as a consequence it has been more difficult to manage and understand.

● Model based has advantages

○ Easier and more transparent way of involving more people in the development, maintenance and testing of new functionality

○ Alignment with other model based standards

○ Enables more straight-forward generation of other expressions of the standard

● New Requirements

○ other implementations of the standard - not just XML but also RDF, program libraries (like JSON, C#, Java), database schemes, etc

○ new functionality e.g. process and automation

○ better documentation of other forms of data collection e.g. clinical, biomedical

○ documenting other forms of data - genetic, big data

○ a better mechanism for sub-setting the standard than instance profiles

**How are we doing it?**

As a self-sustaining membership organization whose members have a voice in the development of the DDI specification, we have made the whole development process as open and transparent as possible.

The project has a website with documentation of the meetings and decisions that take place at [https://ddi-alliance.atlassian.net/wiki/display/DDI4/DDI+Home](https://ddi-alliance.atlassian.net/wiki/display/DDI4/DDI%2BHome%20) and an open development platform where active work on the specification is taking place at<http://lion.ddialliance.org/> (Lion)

The model specification is being created on the Lion development platform. This captures all the objects and their relationships to other object used and the relevant information to support documentation to understand the objects and their intended usage.

**What are the outputs?**

The platform is able to generate the model in both XML Metadata Interchange (XMI) format, and in RDF/OWL and object level documentation in Docbook (reusable) and HTML formats.

The XMI model is processed alongside bindings to generate XML Schema with an XSD file with in-line documentation and an OWL ontology with RDF expressions for implementers. Both have HTML documentation to accompany them.

As a major shift in the way in which a DDI product is developed, the current DDI effort has unfolded over several years in a series of sprints and online meetings. The effort involves developing not only the information model, but also the infrastructure for building that model and then transforming it into a set of representations and associated documentation (initially XML schema and RDF/OWL). The results of the current DDI work will be rolled out in a series of "releases".

The initial releases will be both tests of the production infrastructure and an opportunity for the DDI community to comment on the direction of the information model. In addition to revisions driven by community feedback each of the initial releases will also add new content.

### **III. The Future: DDI-Lifecycle MD (Model-Driven) (from DDI Alliance page)**

In the same way that experiences with DDI-Codebook lead to a broader set of requirements fulfilled by DDI-Lifecycle, implementations of DDI-Lifecycle and the DDI RDF vocabularies have provided new requirements, which will be met by DDI-Lifecycle MD, now under development. DDI-Lifecycle MD will be very different from other DDI work products, because it will be based on an information model of the metadata content. This information model will be available for implementation in standard RDF vocabularies and standard XML structures, which will be equivalent. This form of model-based standard is a best practice in the world of standardization, and several other standards in other domains have been using this approach for many years.

Because the DDI Alliance anticipates that there may be many different implementation technologies using DDI as a basis, having an explicit information model, expressed using the Unified Modeling Language (UML)11, will be of benefit. To provide an example, the Statistical Data and Metadata Exchange (SDMX)12 standard is model-based, and has standard implementations in XML, JSON, and EDIFACT (a flat-file syntax). Additionally, the model was used as the basis for an RDF vocabulary published by the W3C (the home of the base RDF standards themselves).

Thus, moving DDI to a model-based orientation is a way of protecting the standard against technological change, and a way of guaranteeing alignment across different technology implementations. The development and management of the model itself has been done in a way that enables the programmatic generation of derived formats such as the specification representations in XML Schema and OWL/RDF, the documentation, and program libraries. This represents the model-driven DDI-Lifecycle.

As a major shift in the way in which a DDI product is developed, the DDI-Lifecycle MD effort has unfolded over several years in a series of sprints and online meetings. As noted, the effort involves developing not only the information model, but also the infrastructure for building that model, the transformation into a set of representations (initially XML schema and RDF/OWL), and the associated documentation. DDI-Lifecycle MD will be rolled out in a series of “releases”. The initial releases will be both tests of the production infrastructure and an opportunity for the DDI community to comment on the direction of the information model. In addition to revisions driven by community feedback each of the initial releases will also add new content.

DDI-Lifecycle MD will be another type of DDI work product: like DDI-Codebook and DDI-Lifecycle, it will be a standard structure for metadata related to the lifecycle of data in all stages, and thus will encompass much of what is available in those XML structures today. It will also include the functionality of the DDI RDF vocabularies, since these are based on the structures of DDI-Codebook and DDI-Lifecycle. Like DDI-Codebook and DDI-Lifecycle, it will be compatible with the DDI Controlled Vocabulary work products.

However, there will be an expansion of coverage in terms of the data lifecycle (as compared to DDI-Lifecycle 3.\*). For example, DDI-Lifecycle MD supports other types of data collection besides surveys, such as the collection of health information (including Electronic Health Record data), machine data (logs), and other forms of event and administrative data. Support for other data sources has in turn necessitated a review of DDI data processing objects and the development of a free standing process model that is not embedded in survey data collection. This model enables us to represent new types of data lifecycles including those that facilitate the building and maintenance of registries. Registries do not capture data in waves and rounds. Instead they lend themselves to new logical and physical products often supported by NoSQL and other types of big data stores that are able to disaggregate and aggregate data in a wide range of formats as registries evolve over time.

The expansion of coverage doesn’t mean a proliferation of new study objects. Instead a model-based information model often facilitates study object reuse, specialization, and a more systematic approach to study object development and organization.

Likewise, while this may sound like a proliferation of different DDI work products, in fact it is not: once the information model is developed, the XML and RDF specifications will be automatically generated from the model. Thus, although there *will* be an increased number of work products offered by the Alliance to the user community, these will actually require less effort to develop, and will have a guaranteed alignment in terms of their metadata structure.

This supports the use of subsets of the whole model for a variety of purposes. These subsets are called functional views. They are realized in related packages in the model and in the XML and RDF specifications. This approach enables the use of DDI for different audiences, hides complexity of the whole model, and supports interoperability.

Upon review of other documentation, you will find that these functional views are aligned with specific use cases. In the end the goal is that model-based DDI-Lifecycle can present DDI to users in chunks, so users can learn and own just the parts of DDI that they need. Of course, because of their use case(s), some users will necessarily have to become experts. However, the hope is that most users will only need to learn specific views.