Next-Generation DDI: Building Model-Based DDI 4

Mary Vardigan, ICPSR
Wendy Thomas, Minnesota Population Center
Arofan Gregory, Metadata Technology NA
Joachim Wackerow, GESIS

IASSIST 2014
June 4, 2014
About DDI

- A structured metadata specification of and for the community
- Two major development lines – XML Schemas
  - DDI Codebook
  - DDI Lifecycle
- Additional specifications:
  - Controlled vocabularies
  - RDF vocabularies for use with Linked Data
- Managed by the DDI Alliance
  http://www.ddialliance.org
Data Documentation Initiative

Benefits of the DDI Approach

• Rich content
• Metadata reuse across the life cycle
• Machine-actionability
• Data management and curation
• Support for longitudinal data and comparison
Lifecycle Orientation
Accumulating Metadata
Global Reach
Why DDI4?

- Need a stable yet flexible specification that can adapt to new requirements
  - Address needs of new communities
  - Extend lifecycle coverage
  - Improve integration with other standards
High-Level Goals

- Improve DDI usability and accessibility
- Enhance the documentation
- In general, develop a sustainable and transparent approach
Architecture Goals

- Information model as the DDI foundation
- This will enable:
  - Improved communication with other disciplines and standards efforts
  - Flexibility in terms of technical expressions of the model
  - Streamlined development and maintenance
Content Goals

• Abstraction of data capture/collection/source to handle different types of data
• New content on sampling, survey implementation, weighting, and paradata
• New content related to qualitative data
• Framework for access to data and metadata
More Content Goals

• Process description across life cycle, including support for automation and replication
• Integration with GSBPM/GSIM, SDMX, CDISC, Triple-S
• Data management planning
• Fieldwork
How Will We Accomplish This?

- Agile project management
  - Iterative development
  - “Sprints”
- Community involvement
  - Virtual teams
  - Content and data modelers
  - Dedicated project manager
- Support of the Alliance
  - Providing funding for sprints
Activity So Far

- Initial meeting at Dagstuhl training center in 2012 to set requirements and design principles
- A paper resulted on DDI Moving Forward: [http://dx.doi.org/10.3886/DDIWorkingPaper04](http://dx.doi.org/10.3886/DDIWorkingPaper04)
Design Principles

1. Interoperability, Standards
2. Simplicity
3. User Driven
4. Terminology
5. Iterative
6. Documentation
7. Lifecycle Orientation
8. Reuse and Exchange
9. Modularity
10. Stability
11. Extensibility
12. Tool Independence
13. Innovation
14. Actionable Metadata
Sprints

• Sprints were held at these locations:
  – Wadern, Germany (Dagstuhl), October 2013
  – EDDI (Paris), December 2013
  – NADDI (Vancouver), April 2014
  – IASSIST (Toronto), June 2014

• Next sprint: Dagstuhl, October 2014
# Data Documentation Initiative

## Release Plan (subject to change)

<table>
<thead>
<tr>
<th>Stage 1</th>
<th>Modelling Completed</th>
<th>Production/ Documentation and TC Review completed</th>
<th>Out for community comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent (Library)</td>
<td>Start of June 2014</td>
<td>Mid September 2014</td>
<td>End September 2014</td>
</tr>
<tr>
<td>Core (Library)</td>
<td>All modelling completed and reviewed at IASSIST Sprint.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conceptual (Library)</td>
<td></td>
<td>(Note: Longer period given for first run through this process to give time to sort out bugs)</td>
<td></td>
</tr>
<tr>
<td>Process (Library)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discovery (View)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage 2</th>
<th>Modelling Completed</th>
<th>Production/ Documentation and TC Review completed</th>
<th>Out for community comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Instrument (View)</td>
<td>End of September 2014</td>
<td>End of November 2014</td>
<td>Mid December 2014</td>
</tr>
<tr>
<td>Simple Data Description (View)</td>
<td>All modelling completed and reviewed via virtual teams over European summer months</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classifications (View)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage 3</th>
<th>Modelling Completed</th>
<th>Production/ Documentation and TC Review completed</th>
<th>Out for community comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Codebook (Composite View)</td>
<td>End of January 2015</td>
<td>Mid March 2015</td>
<td>End March 2015</td>
</tr>
<tr>
<td>Comparison/ Harmonisation (View)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Methodology (View)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complex Instrument (View)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complex Data Description (View)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To be Determined</th>
<th>Modelling Completed</th>
<th>Production/ Documentation and TC Review completed</th>
<th>Out for community comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection Management (View)*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Study Inception (View) and Data Management Plan (View)*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Production (View)**</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualitative (View)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Survey Development (View)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fieldwork (View)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

These views will be scoped at Sprints as possible. Any modelling work to be done on these views is subject to the relevant user groups providing resourcing. If resourcing is provided (in the form of money or people), then the work will be prioritised to an earlier release.
We need you!

• Please join our effort
• There are many ways to contribute:
  – Content modeling
  – Data modeling
  – Use cases
  – Production
  – Review of output
• Contact Therese Lalor or Mary Vardigan
Next Generation DDI: Content
Object Library vs. Functional Views

- All DDI objects will be in a central library
- Objects will be organized within the library to facilitate management
- Users will primarily see views that are specific to certain activities or information areas
  - Questionnaire description
  - Simple Data File
Imagine the DDI4 is like an archive. There is a room full of the information that DDI 4 covers. This information in is nicely organised into rows and shelves and boxes.
Typical users will come to the archive with a particular use case in mind. The archivist will give them the subset of information that is most useful to them.
Sophisticated users will be able to go into the actual archive to find the set of information that is most useful to them. Most people never see the way the information is sorted into rows, shelves and boxes.
In DDI 4 terms

Library

Data sets
Physical data structures
Logical data structures
Package
Study
Glue
Constructs
In DDI 4 terms
Creating DDI 4

We are in the process of building the library. The plan is to put in the information that we think the most people will need first.
Creating DDI 4

We know this is not complete and don’t know exactly how best to organise the information yet.
Creating DDI 4

We will start to work with people who have particular use cases (e.g. describing a simple instrument). We will have some of the information that is important to them and they will help us find the bit we are missing…and so the library get more things in it.
Creating DDI 4

The further we go through the process, the less new things should have to be added to the library. For example, there should already be a lot of information in the library for complex instrument, because we already have the information for simple instrument.

Each release builds on the information already in the library.
Development of DDI Content

• We have loaded all of the 3.2 content to make it available for use within DDI 4

• Content groups will come together to define specific views
  – What is the coverage
  – What objects are needed
  – What objects already exist
  – What objects need to be created
<table>
<thead>
<tr>
<th>TITLE</th>
<th>PURPOSE</th>
<th>DESCRIPTION OF VIEW</th>
<th>STATUS</th>
<th>PROPOSED BY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Management Plan</td>
<td>Data management plan; development, content, execution</td>
<td></td>
<td>PROPOSED</td>
<td>Moving Forward Project</td>
</tr>
<tr>
<td>Simple Data Description</td>
<td>To develop a robust model that can describe all aspects of simple, rectangular data file in our domain. The model must include bridges from the physical representation of a rectangular data file to high-level conceptual objects in the model.</td>
<td></td>
<td>ACCEPTED RELEASE 2</td>
<td>Moving Forward Project</td>
</tr>
<tr>
<td>Comparison/Harmonization</td>
<td>aka Concordance processing and products.</td>
<td></td>
<td>ACCEPTED RELEASE 3</td>
<td>Moving Forward Project</td>
</tr>
<tr>
<td>Analysis and dissemination</td>
<td>This view describes the activities involved in the analysis of final data products produced from data collection, and the dissemination of the findings of that analysis to the relevant user community.</td>
<td>This includes describes the implementation of analytical procedures applied to data files for the purposes of answering research questions or business needs specified in the study inception. The design of these procedures is specified in the project methodology….</td>
<td>IDEA</td>
<td>Steve McEachern</td>
</tr>
</tbody>
</table>
Interactive work with modelers

- Each content group has a team leader, a modeler and a number of members who define, refine and comment on the content
- Modelers will make decisions regarding the internal structure of the object library
- Content groups will define what should be in a specific view
<table>
<thead>
<tr>
<th>Backlog</th>
<th>To Do Priority</th>
<th>In Progress</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Content:</strong></td>
<td><strong>Modeling:</strong></td>
<td><strong>Content Team Tasks:</strong></td>
</tr>
<tr>
<td>• Field work management</td>
<td>• Administrative metadata</td>
<td><strong>Scope View</strong></td>
</tr>
<tr>
<td>• Qualitative</td>
<td>• Grouping</td>
<td>• Classification</td>
</tr>
<tr>
<td>• Classification</td>
<td>• Containership and reference</td>
<td>• Study Inception</td>
</tr>
<tr>
<td>• Survey development</td>
<td></td>
<td>• Data Management Plan</td>
</tr>
<tr>
<td>• Complex survey</td>
<td></td>
<td>• Collection Management</td>
</tr>
<tr>
<td>• Data capture methodology</td>
<td></td>
<td>ID objects</td>
</tr>
<tr>
<td>• Comparison / Harmonization</td>
<td></td>
<td>• Simple Data Description</td>
</tr>
<tr>
<td>• Complex data description</td>
<td></td>
<td>• Simple Codebook</td>
</tr>
<tr>
<td>• Complex survey</td>
<td></td>
<td>• Simple Instrument</td>
</tr>
<tr>
<td>• Data production</td>
<td></td>
<td>Capture Content in Drupal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Primitives</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Extended Primitives</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Basic reusable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Agent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Discovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Conceptual</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Object QA</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Modeling integration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sync back to Drupal</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>User Documentation</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Modeling Team Task:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Integrating other standards</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Web services</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Identification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Name, Label, Description</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Infrastructure Tasks:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Domain</td>
<td>Details</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Core</td>
<td>Rules for primitives and extending primitives, Drupal updated</td>
<td></td>
</tr>
<tr>
<td>Agent</td>
<td>All objects in Drupal and reviewed, Update for team</td>
<td></td>
</tr>
<tr>
<td>Conceptual</td>
<td>All objects in Drupal and reviewed, Update for team</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>All objects in Drupal and reviewed, Update for team</td>
<td></td>
</tr>
<tr>
<td>Discovery</td>
<td>New objects created in Drupal, view created and reviewed, Update for team</td>
<td></td>
</tr>
</tbody>
</table>
Next Generation DDI: Model-Driven Technical Framework
What is the DDI 4 Model?

• The DDI 4 model will be formally expressed as a UML model
  – Metadata objects are expressed as UML classes
  – The model defines each object
  – It shows the exact relationships between them
  – Describes all object properties
  – A restricted (and documented) set of UML features is used

• Unlike some conceptual or reference models, DDI 4 is an implementation model
  – Could be used to implement the GSIM model, for example
  – Mappings to other models will be formally documented for each object
What Does It Look Like?
The Model - Organization

• The “complete” DDI 4 is a library of objects, organized thematically

• Each function to be supported by DDI is expressed as a selection of related objects called a “Functional View”
  – This is just a subset of the model
  – Designed to meet specific requirements
  – A DDI Codebook would be represented as a functional view, for example (to support migration)

• Examples: Simple Data Description View, Discovery Metadata View, Agent View (individuals, organizations, and “machines”)
Data Documentation Initiative

Model

Workbench

DDI 3.2 (as incorporated by content teams)

Draft View 1

Draft View 2

From Drupal

Library

Core

Study

Data

Functional Views

Final View 1

Final View 2

Primitives

Ext. Primitives

Objects

Objects

Objects

Objects

Publication
Model Developers and Users

- The DDI Alliance publishes both the library and a standard set of functional views
  - The views are part of the model
- Developers work in an unpublished “workbench” portion of the model
  - The Data Modelling Team organizes and quality assures the library and the views for publication
- Sophisticated users can create their own views (for custom functions)
  - For example, a community of users like CESSDA might develop a functional view
  - This could be submitted for possible inclusion in the official DDI model
Design Principles – High Level

- **Interoperability and Standards** – The model is optimized to facilitate interoperability with other relevant standards.
- **Simplicity** – The model is as simple as possible and easily understandable by different stakeholders.
- **User Driven** – User perspectives inform the model to ensure that it meets the needs of the international DDI user community.
- **Terminology** – The model uses clear terminology and when possible, uses existing terms and definitions.
- **Iterative Development** – The model is developed iteratively, bringing in a range of views from the user community.
- **Documentation** – The model includes and is supplemented by robust and accessible documentation.
- **Lifecycle Orientation** – The model supports the full research data lifecycle and the statistical production process, facilitating replication and the scientific method.
- **Reuse and Exchange** – The model supports the reuse, exchange, and sharing of data and metadata within and among institutions.
- **Modularity** – The model is modular and these modules can be used independently.
- **Stability** – The model is stable and new versions are developed in a controlled manner.
- **Extensibility** – The model has a common core and is extensible.
- **Tool Independence** – The model is not dependent on any specific IT setting or tool.
- **Innovation** – The model supports both current and new ways of documenting, producing, and using data and leverages modern technologies.
- **Actionable Metadata** – The model provides actionable metadata that can be used to drive production and data collection processes.
Supporting Design Principles (Some Examples)

- **Interoperability and Standards** – The model is optimized to facilitate interoperability with other relevant standards.
  - Model captures formal links with other models and standards
- **Simplicity** – The model is as simple as possible and easily understandable by different stakeholders.
  - The modeling style is restricted and simple
  - The Functional Views are simple subsets
- **Modularity** – The model is modular and these modules can be used independently.
  - The library is organized into themed UML packages
Supporting Design Principles (cont)

• **Stability** – The model is stable and new versions are developed in a controlled manner.
  – No published object will ever be deleted from the model
  – Objects and Views will be versioned

• **Extensibility** – The model has a common core and is extensible.
  – Sophisticated users can extend the library
  – They can also use our production framework

• **User Driven** – User perspectives inform the model to ensure that it meets the needs of the international DDI user community.
  – Sophisticated user communities can make their views available to others
DDI 4 Products

• Whole Model:
  – A PDF specification of the entirety of DDI 4. (Needs high-level documentation)
  – An EA file containing DDI 4 (also the XMI)
  – HTML documentation for DDI 4 (Needs high-level documentation)

• For each Functional View:
  – PDF specification of the view (Needs high-level doc)
  – XSD for the View (with field-level documentation inline)
  – Clickable XSD documentation (created using the same tool as other versions of DDI-Lifecycle)
  – View OWL (in RDF XML) - field level doc as comments
  – View HTML documentation for OWL
  – View model documentation in HTML

• Tools:
  – The production framework itself as a toolkit for those making their own views.

• Other non-XSD, non-RDF deliverables are being considered (SQL, Jason, Java, C#, etc.)
Creating DDI Products (with other standards!)

- RDF and XSD products are created automatically using XSLT transforms
  - The model acts as the basis for these products
  - We are using XMI, a standard for describing UML models
- Documentation is using DocBook XML
  - XSLT and XSL-FO are used to create documentation in HTML and PDF
  - Most detailed documentation is contained in the model itself
  - Some high-level documentation is created for the purposes of having coherency
Content modelers define requirements for views and create needed objects.

Corrections are made by both teams working together.

Data modelers do integration, validation, quality assurance.

Content capture

UML class model

XMI

EA

pictures

text, model

XSD

OWL

Git

Java, SQL...

HTML

PDF

Docbook

Drupal

High level text

Documentation team creates high level documentation
# Production Tools, Output, and Instances

<table>
<thead>
<tr>
<th>Instances (of XML Schema and OWL/RDF representations)</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Representations (of subsets of model) in XML Schema and OWL/RDF</td>
<td>Specification</td>
</tr>
<tr>
<td>Functional Views (subsets of model)</td>
<td>Development</td>
</tr>
<tr>
<td>Object Library (model in EA)</td>
<td></td>
</tr>
<tr>
<td>Workbench (in EA)</td>
<td></td>
</tr>
<tr>
<td>Drupal</td>
<td></td>
</tr>
</tbody>
</table>
XML Schemas

- For each standard view there will be an XML document type
  - It will be focused on performing a specific function
  - Each object in the model becomes an XML element
  - Each property becomes either:
    - An attribute (for simple content)
    - An element holding attributes for special cases where complex content is treated as simple content in the model
XML Schemas (cont.)

• The XML will be very consistent
  – With the model
  – Within itself – it is auto-generated

• There will be only one place where a specific type of metadata can be found in any XML document
  – Developers insisted on this

• Most of the “packaging” elements will be removed
  – It is much less deeply nested than DDI Lifecycle is today
Example XML

```xml
<DataDescriptionDocument>  
<DataDescription>  
<VariableDescriptionReference/>  
<VariableDescriptionReference/>  
<Purpose/>  
</DataDescription>  
<VariableCollection>  
<VariableDescription/>  
<VariableDescription>  
<CodeListRef id="xxxx"/>  
</VariableDescription>  
</VariableCollection>  
<CodeListCollection>  
<CodeList/>  
</CodeListCollection>  
</DataDescriptionDocument>  
<CodeList id="xxxx" label=""/>  
<Code/>  
<Code/>  
</CodeList>
```
RDF Deliverables

- For each standard view there will be an RDF vocabulary published
  - Described using OWL, expressed in RDF XML
- The RDF will correspond directly to the model
  - Relationships are expressed in RDF
  - Complex properties are expressed in RDF
  - Simple properties are RDF literals
- Some objects will be mapped into their equivalents in other common vocabularies (e.g., foaf) using the OWL functions which support this
<owl:Class rdf:about="http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Agents#Agent">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:isDefinedBy rdf:resource="http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Agents"/>
    <rdfs:label xml:lang="en">agent (Abstract)</rdfs:label>
</owl:Class>
Discovery

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#Category

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#CategoryStatisticType

Sub Class of: http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/ExtendedPrimitives#StatisticType

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#CodedGeography

Sub Class of: http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#SpatialCoverage

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#Coverage

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#ExternalGISSystem

Sub Class of: http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#SpatialCoverage

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#ExternalTopic

Sub Class of: http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#TopicalCoverage

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#GeographicCodeList

Sub Class of: http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#SpatialCoverage

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#Keyword

Sub Class of: http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#TopicalCoverage

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#SpatialCoverage

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#Subject

Sub Class of: http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#TopicalCoverage

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#SummaryStatisticType

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#TemporalCoverage

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#SpatialCoverage

Sub Class of: http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#TopicalCoverage

Class: disco:http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Discovery#TopicalCoverage

Properties

Datatype Property: http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Antecedents#effectivePeriod
(Domain: disco:Agents#AdditionalInformationType -> Range: # )

Datatype Property: http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Antecedents#privacy
(Domain: disco:Agents#AdditionalInformationType -> Range: # )

Datatype Property: http://rdf-vocabulary.ddialliance.org/DDICoreVocabulary/Antecedents#cityPlaceLocal
Syntax Outputs

• The model will retain consistency with DDI Lifecycle 3.*
  – Many objects were based on imported DDI – Lifecycle version 3.2 elements
  – There is a “no gratuitous re-modelling” rule

• Metadata will be round-trippable between RDF and XML expressions
  – This has implications for how identification is done
What about earlier versions?

- Earlier versions will continue to be supported
  - DDI-L (3.0, 3.1, 3.2)
  - DDI-C
- Many users will continue to use the earlier versions
- Migration paths from 3.2 and 2.5 will be available
- Functional views may support migration
  - There could be a DDI 2.5 functional view
  - There could be a functional view covering Disco or XKOS
Conclusion

• DDI 4 will represent a major step forward in how well we can meet the needs of our users
• It will offer a better technical foundation for implementation
  – The model provides a strong basis for using different technology approaches
• It will be better documented and more approachable
  – Functional Views are simpler to understand than monolithic standards
  – The model provides a strong set of reusable documentation
• It will be more stable, and yet flexible enough to meet the emerging needs of users over time
Want to know more about DDI 4.0?

- Minutes and formal products from the Dagstuhl and EDDI sprints are found at the DDI site
  - http://www.ddialliance.org/ddi-moving-forward-process
- A new public wiki has been established to maintain and organize all working and final documents
  - http://www1.unece.org/stat/platform/display/DDI4/DDI+Home
- Current development platform for content capture
  - http://lion.ddialliance.org/
Involvement

- Participate in a Sprint
- Provide use cases for content groups
- Volunteer to be on a virtual team
  - Conference calls
  - Review and comment on content
  - Define or refine objects in Drupal
  - Write documentation
- Review and comment on the specifications when published as beta’s or through the formal review process