



DDI Moving Forward

Update, Feedback, and
Suggestions

NADDI 2015
Madison, Wisconsin

Arofan Gregory, Wendy Thomas,
and Mary Vardigan

Presentation Outline

- Overview – Mary
- Content and development process -- Wendy
- Technical perspective – Arofan
- Discussion -- All



Background

- DDI Alliance building a model-driven specification
- Goals
 - Extend Lifecycle with new content
 - Expand domain coverage of DDI
 - Render the specification in new ways
 - Reduce complexity of DDI
 - Develop DDI more sustainably



Activity: Events

- 2012 – Foundational Dagstuhl Seminar
- 2013 – Sprint 1 at Dagstuhl
- 2013 – Sprint 2 at EDDI in Paris
- 2014 – Sprint 3 at NADDI in Vancouver
- 2014 – Sprint 4 at IASSIST in Toronto
- 2014 – Sprint 5 at Dagstuhl (included citation work)
- 2014 – Sprint 6 at EDDI in London
- 2015 – Sprint 7 at IASSIST in Minneapolis



Activity: Virtual Meetings

- Task Teams meet weekly or biweekly:
 - Classification View Team
 - Modelling Team
 - Simple Codebook View Team
 - Simple Data Description View Team
 - Simple Instrument View Team
 - Tools Support Team



Governance

- Advisory Group oversees development
 - Adam Brown, Statistics New Zealand, Chair
 - Dan Gillman, US Bureau of Labor Statistics
 - Arofan Gregory, Metadata Technology
 - Larry Hoyle, University of Kansas
 - Steve McEachern, Australian Data Archive
 - Wendy Thomas, Minnesota Population Center
 - Mary Vardigan, ICPSR, University of Michigan
 - Joachim Wackerow, GESIS – Leibniz Institute for the Social Sciences
- Executive Board gets periodic updates
- Project Coordinator needed!



Content and Development Process

- Content coverage
 - Library Packages
 - Functional Views
- Laying the building blocks
- “Simple to complex” – one way to approach complexity
- Road map to completion
- Developing Functional Views



Content Coverage

- Commitment to cover everything in DDI-Lifecycle
- Expansion to include:
 - Processing model
 - Qualitative data
 - Broader range of quantitative data capture activities
 - Register data
 - Bio-markers
 - Event data



Library Packages and Functional View

- Library Package
 - a group of classes within the Library related by a theme. The Class Library for DDI 4 encompasses the entire DDI 4.0 model, but without any specific schemas or vocabularies for Functional Views. Classes contain primitives and extended primitives and are the building blocks used to construct the Functional Views.
- Functional View
 - a set of classes that are needed to perform a specific task. It primarily consists of a set of references to specific versions of classes. Functional Views are the method used to restrict the portions of the model that are used, and as such they function very much like DDI profiles in DDI 3.*.



Laying the Building Blocks

- The content of Q1 focuses on the building blocks needed to provide the foundation for more complex coverage
- Primitives and Complex Data Types
 - A consistent means of describing multi-language string
 - Commonly uses constructs such as Label, Description, and Definition
- Standard structures
 - Base for Identifiable classes
 - How to create collections (groups of classes)
 - Process model



Conceptual Base of Data

- Concepts, Objects, and enumeration
 - How concepts and objects are described
 - How a concept is enumerated in relationship to a specific object
- Conceptual Variable, Represented Variable, and Instance Variable
 - Moving from abstraction to application
 - Important for supporting longitudinal studies and intended comparability



“Simple to Complex”

- We start with the basics
 - Everyone needs to describe things for understanding
 - There are basic requirement needed to understand data
- We move to the complex
 - Some need to describe additional detail
 - Some need to prescribe activities in a machine actionable way
- By building up from same basic structures we recognize the commonality of data from a wide array of sources



Roadmap

Out for Review	Content for Release
Q1 2015	Agent (Library Package), Agent (View), ComplexDataTypes (Library Package), Conceptual (Library Package), Core Process (Library Package), Discovery (Library Package), Discovery (View), Identification (Library Package), Primitives (Library Package), Representations (LibraryPackage) , Utility (Library Package)
Q2 2015	Classification (View), Simple Instrument (View), Simple Data Description (View) , Data Capture (Library Package), HistoricalProcess (LibraryPackage), PrescriptiveProcess (LibraryPackage)
Q3 2015	Methodology , Simple Codebook , Enhanced Citation , Complex Instrument , Complex Data Description
As supported	Data Management Plan , Comparison/Harmonization , Study Inception , Data Collection and Processing , Collection Management



Developing Functional Views

- Functional Views reflect the needs of sets of related Use Cases
 - Basic (or Simple) Codebook
 - Questionnaire Design
 - Classification Management
 - Managing Organizations, Individuals, and other Agents
- Working groups are developed around sets of Use Cases
- Working groups determine what metadata is needed
 - Select from existing classes for inclusion
 - Draft new classes or extension of an existing class to meet the needs of the Use Case



Modelers Group and Technical Committee

- Members of the Modelers Group work with the groups designing Functional Views to assist in design and identification of appropriate classes
- Modelers Group is responsible for
 - Developing modeling rules
 - Ensuring smooth integration into the model as a whole
 - Adherence to modeling rules
 - Resolving issues between and within functional groups
- Technical Committee ensures cohesion with earlier DDI specification and prepares the materials provided by the Modelers Group for review
 - Manages the review process and disposition of comments



Why a Model-Driven Standard?

- There are many different ways of implementing DDI:
 - As XML
 - As RDF
 - In application languages (Java, C#, etc.)
 - In relational databases
 - In other types of databases/languages (JSON, Hadoop, etc.)
- A model is “neutral” to any of these
- A model can be expressed as any of these
- This is current best practice for standards development
 - Makes the standard easier to understand and implement
 - Emphasizes complete documentation of the classes in the model



Production Process

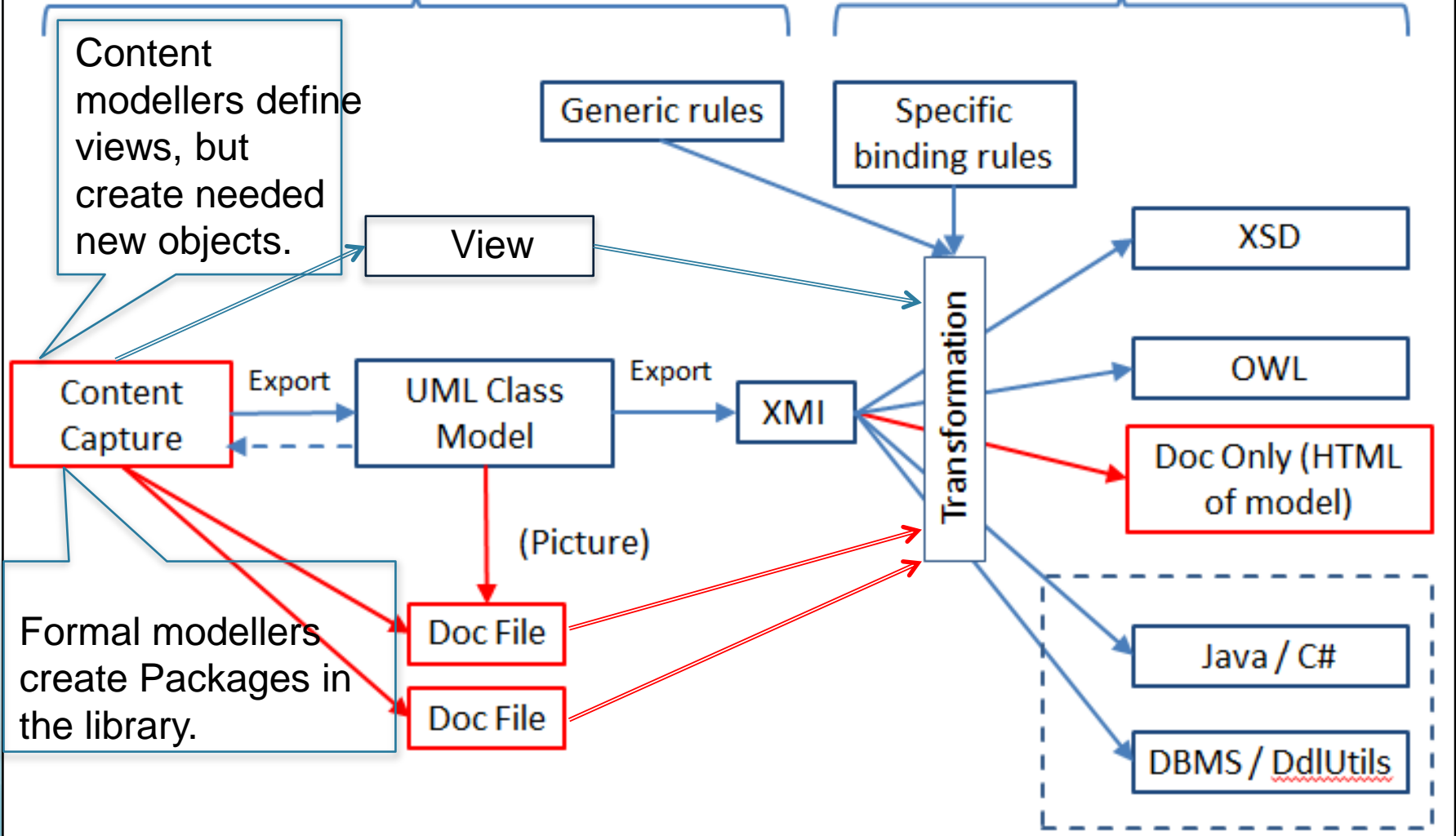
- The model is a simple profile of the UML notation for formal models
 - It has complete documentation and definition of all classes
 - It shows all properties and relationships of classes
- Once complete, the model uses automated transformations to
 - Create end-user documentation
 - Create XML schemas
 - Create OWL declarations for RDF vocabularies



Framework

Platform independent

Platform specific



The DDI Modeling Approach

- Powerful, generic modeling “patterns” are used to ensure consistency of modeling
 - Process model (for historical and prescriptive models, instrument flows)
 - Collection model (for all types of grouping)
 - Customization meta-model (for all types of self-describing metadata)
- Generic models are refined for specific uses
- Everything in the model must be “real”
 - Absence of packaging, unlike DDI 3.*



DDI 4 XML

- The style of DDI XML will be very different
 - Less deeply nested
 - Only one place for any given type of “class” (XML element) to be found in any given XML schema
 - Very consistent due to auto-generation from the model
- Each Functional View is its own XML document
- There will be an “uber-view” – an XML schema for the entire library
- The XML bindings from the model are now being tested in earnest as of Q1 Review
 - Very likely to need some adjustments



DDI 4 RDF Vocabularies

- Each functional view will be packaged as an RDF vocabulary described using OWL
- All classes, relationships, and properties will follow the DDI 4 model exactly
 - Some classes/relationships/properties will be mapped to common pre-existing RDF vocabularies (FoaF, DCAT, etc.)
- As for the XML schemas, the binding to RDF is only now being seriously tested, and may need adjustment (as of Q1 Review)



Other Technical Goodness

- Standard web services interfaces for DDI applications
 - For generic querying and retrieval of DDI metadata in XML form
- New possibilities for customized metadata
 - Will be described and exchanged using meta-model structures (similar to SDMX and GSIM)
 - “Custom” metadata can be made interoperable
- Strong alignment of model with other relevant models/vocabularies
 - Generic Statistical Information Model (GSIM)



Thanks!

