The DDI4 Prototype

Jay Greenfield, Ph.D.
and
The DDI4 Modeling Team
(Dan Gillman, Jay Greenfield, Larry Hoyle and Wendy Thomas)

April 12 2018
What is new in DDI4?

1. Packaging and the user experience
2. Support for new data sources beyond questionnaires
3. Support for new types of data collections beyond unit data records and cubes
4. A process model able to describe both business workflows and data capture workflows retrospectively and prospectively

• Retrospective and prospective uses of the process model are being tested with several platforms including Statistical Packages like SAS, R and Stata, Extract Transform and Load (ETL) systems and Business Process Model and Notation (BPMN) systems
What’s new in DDI4: Packaging

• The **new package** is NOT massive anymore. It comes in **views**
  • The development of views has been use-case driven
  • Not to be exhaustive, here are a few:
    • **StatisticalClassificationView** (At once GSIM-compatible and GSIM-extended)
    • **DescriptiveCodebookView** (Nesstar friendly)
    • **DataManagementView** (Crazy new!)
    • **DataDescriptionView** (Supports both the logical and physical description of just about any data type you can imagine)
    • **DataCaptureInstrumentView** (Not questionnaire-centric anymore)
What’s new in DDI4: Packaging (continued)?

• The **new package** is NOT massive anymore. It comes in views...
  • Views restrict relationships between classes from the larger model
    • Between views there are boundaries and not very much bleed
    • Users can engage multiple views as needed to tell their data stories
  • Each view comes in two favors – **XML** and **RDF**
  • That means metadata developers (human and/or software agents) can create both XML and RDF instances of each view
  • Validation support is available for both the XML (XSDs) and RDF (**ShEx**) instances
What is new in DDI4: Support for New Data Sources

• Today we collect data from registries, electronic health records (EHRs) and social media as well as questionnaires

• In this context, an “instrument” is not always a questionnaire

• Instead an instrument needs to be the traversal of any data source
  • Across this traversal we perform capture
  • In a capture we might ask questions or take measurements
  • Measurements work with the new data sources
  • Now with an instrument, using measurements, we can capture registry, EHR and social media data
What is new in DDI4: Support for New Data Collections (1 of 5)

• GSIM had the idea that a data collection (unit or aggregate) consists of measurements, attributes and identifiers

• GSIM conflates these components of a data collection with its structure

• DDI4, on the other hand, takes the view that the structure of a data collection is quite different
  • In the simple case, a data collection is an ordered list of instance variables
  • In the structured case, a data collection is a graph...
What is new in DDI4: Support for New Data Collections (2 of 5)

• Let’s drill down into the DDI4 graph data collection
  • The implementation is **standard**: graphs may be represented as a list of adjacency lists
  • _Here for each vertex we create an array of vertices adjacent to it_
  • Now imagine each vertex is an Instance Variable
  • In DDI4 we call this thing on the right an **InstanceVariableRelationStructure** (IVRS)
  • In the context of **ISO-11404**, the InstanceVariableRelationStructure (IVRS) is a **data types factory**
What is new in DDI4: Support for New Data Collections (3 of 5)

• What can we do with the factory?
  • The graph on the right could be used to describe a social network where each instance variable is an identifier
  • Likewise it could be used to describe a medical information model where the IVs are groups of measures and attributes:
What is new in DDI4: Support for New Data Collections (4 of 5)

• What can we do with the factory?
  • More generically, we can use it to describe **ANY business object**:

  ![JSON objects diagram]

  - Note that a complete description of a JSON object in DDI4 starts with an InstanceVariableRelationStructure.
  - This IVRS, in turn, has a corresponding layout or, again, physical format the IVRS passes through to produce actual JSON.
What is new in DDI4: Support for New Data Collections (5 of 5)

- What can we do with the factory?
  - And just for fun, we can use it to describe a region of the *Google Knowledge Graph*...
What is new in DDI4: A DataPipeline that traverses a Business Process Model

Study

DataPipeline

BusinessProcess+

BP AlgorithmOverview

Step*

Precondition*

LogicalRecord*

Postcondition*

LogicalRecord*

StandardModelUsed

GSBPM/GLBPM Step

Domain-specific Specialization
What is new in DDI4: A DataPipeline that traverses a Business Process Model

Currently, the DataPipeline is a Simple Collection (ordered list) of BusinessProcesses

The DDI4 Prototype

Study
What is new in DDI4:

A DataPipeline that traverses a Business Process Model

Each BusinessProcess is associated with a generic business process model and a domain-specific specialization.
Each BusinessProcess has an AlgorithmOverview which at a high level describe a sequence of workflow steps.

Optionally, each BusinessProcess has a WorkflowStepSequence or a WorkflowStepRelationStructure. These are respectively Simple and Structured Collections of WorkflowSteps.

Sometimes we omit the specification of WorkflowSteps because, for a certain audience, we want to keep the description at a high level.
What is new in DDI4: A DataPipeline that traverses a Business Process Model

A BusinessProcess has zero or more Preconditions and Postconditions

Preconditions and Postconditions are LogicalRecords that in DDI4 both UnitDataRecords and AggregateDataRecords specialize

Here the Preconditions for this BusinessProcess are the Postconditions from the previous one: the set of staging tables that were created to host the visit data.

The Postcondition is a loaded RawMicroData file.

```xml
<PreCondition>
  <UsesLogicalRecord typeOfClass="UnitDataRecord">UnitRecIndividualEvents</UsesLogicalRecord>
  <Sql>CREATE TABLE ${SiteCode}_IndividualEvents ( IndividualId VARCHAR(128)

<PreCondition>
  <UsesLogicalRecord typeOfClass="UnitDataRecord">UnitRecIndividualsID</UsesLogicalRecord>
  <Sql>CREATE TABLE ${SiteCode}_Individuals ( IndividualId VARCHAR(128)

<PreCondition>
  <UsesLogicalRecord typeOfClass="UnitDataRecord">UnitRecDeliveriesID</UsesLogicalRecord>
  <Sql>CREATE TABLE ${SiteCode}_Deliveries ( MotherId VARCHAR(128)

<PreCondition>
  <UsesLogicalRecord typeOfClass="UnitDataRecord">UnitRecChildDeliveriesID</UsesLogicalRecord>
  <Sql>CREATE TABLE ${SiteCode}_ChildDeliveries ( childId VARCHAR(128)

<PreCondition>
  <UsesLogicalRecord typeOfClass="UnitDataRecord">UnitRecDeathsID</UsesLogicalRecord>
  <Sql>CREATE TABLE ${SiteCode}_Deaths ( IndividualId VARCHAR(128), Cause1 VARCHAR(40), Cause2 VARCHAR(40), Cause3 VARCHAR(40)

<PostCondition>
  <UsesLogicalRecord typeOfClass="UnitDataRecord">UnitRecRawMicroDataID</UsesLogicalRecord>
  <Sql>CREATE TABLE ${SiteCode}_RawMicroData ( RecNr INT, CountryId INT, CentreId VARCHAR(5), IndividualId VARCHAR(128), Sex INT, </PostCondition>
```
What is new in DDI4: A DataPipeline that traverses a Business Process Model

The record definitions for precondition and postcondition LogicalRecords are listed in the study DataStore and subsequently specified InstanceVariable by InstanceVariable

The DDI4 Prototype
What is new in DDI4: The Business Workflow

• In the last few slides we have been seeing snippets of an XML instance of the DataManagementView
  • The entire instance can be found at DDI4_ALPHA_6.1_DataManagementView
  • This instance was generated by a program that reads a data management platform together with a metadata infusion file created by data scientists and domain experts
  • The story of the instance is that ALPHA (Analyzing Longitudinal Population-based HIV/AIDS data on Africa) is migrating its data management activities across the 10 sites that compose the NETWORK away from statistical packages like Stata, SAS and R to an open source data integration platform called Pentaho
  • The migration is funded by the Wellcome Trust and the Gates Foundation
What is new in DDI4: The Business Workflow

In the last few slides we have been seeing snippets of an XML instance of the DataManagementView.
The entire instance can be found at DDI4_ALPHA_6.1_DataManagementView.
This instance was generated by a program that reads a data management platform together with a metadata infusion file created by data scientists and domain experts.
The story of the instance is that ALPHA (Analyzing Longitudinal Population-based HIV/AIDS data on Africa) is migrating its data management activities across the 10 sites that compose the NETWORK away from statistical packages like Stata, SAS and R to an open source data integration platform called Pentaho.
The migration is funded by the Wellcome Trust and the Gates Foundation.
What is new in DDI4: The Business Workflow (continued)

• The current state of the DataManagementView is that it is being field tested with ALPHA

• The DataManagementView has an executive level description of a DataPipeline, much as we have just seen. Additionally...
  • Each BusinessProcess optionally has a WorkflowStepSequence or a WorkflowStepRelationStructure through which the BusinessProcess can be broken out into a collection of WorkflowSteps
  • Recall that a WorkflowStepRelationStructure, like an InstanceVariableRelationStructure, is a graph
What is new in DDI4: The Business Workflow (continued)

The current state of the DataManagementView is that it is being field tested with ALPHA.

The DataManagementView has an executive level description of a DataPipeline, much as we have just seen. Additionally...

Each BusinessProcess optionally has a WorkflowStepSequence or a WorkflowStepRelationStructure through which the BusinessProcess can be broken out into a collection of WorkflowSteps.

Recall that a WorkflowStepRelationStructure, like an InstanceVariableRelationStructure, is a graph. Here a non-linear collection of WorkflowSteps (WorkflowStepRelationStructure) is first rendered as GraphML and then read by an open source GraphML reader.
What is new in DDI4: The Business Workflow (continued)

- The current state of the DataManagementView is that it is being field tested with ALPHA.
- The DataManagementView has an executive level description of a DataPipeline, much as we have just seen. Additionally...
- Each BusinessProcess optionally has a WorkflowStepSequence or a WorkflowStepRelationStructure through which the BusinessProcess can be broken out into a collection of WorkflowSteps.
- Recall that a WorkflowStepRelationStructure, like an InstanceVariableRelationStructure, is a graph here a non-linear collection of WorkflowSteps (WorkflowStepRelationStructure) is first rendered as GraphML and then read by an open source GraphML reader.

Note that GraphML is an input format supported by many BPMN editors and platforms so...
What is new in DDI4: Process Decomposition

WorkflowStep Collection

WorkflowStep -> Control Construct

- Computation
  - Action

- Metadata Driven
  - Action

  - Instrument
    - Component

  - Question

  - Measurement
What is new in DDI4: Process Decomposition

Both a **BusinessProcess** and a **ConceptualInstrument** are associated with either a Simple Collection of WorkflowSteps (WorkflowStepSequence) or a Structured Collection of WorkflowSteps (WorkflowStepRelationStructure)
What is new in DDI4: Process Decomposition

ComputationActions describe coded-based WorkflowSteps in a WorkflowStepSequence or WorkflowStepRelationStructure.
What is new in DDI4: Process Decomposition

ComputationActions provide an extensible framework of CommandCode for specific computation or transformation objects.

The C²Metadata Project funded by the National Science Foundation produces DDI3/DDI4 CommandCode from SPSS, SAS, Stata and R command scripts.

ComputationActions describe coded-based WorkflowSteps in a WorkflowStepSequence or WorkflowStepRelationStructure.

ComputationAction

Metadata Driven Action

Instrument Component

Question

Measurement
What is new in DDI4: Process Decomposition

The DDI4 Prototype

WorkflowStep Collection

ComputationActions provide an extensible framework of CommandCode for specific computation or transformation objects. ComputationActions describe coded-based WorkflowSteps in a WorkflowStepSequence or WorkflowStepRelationStructure.

The C²Metadata Project funded by the National Science Foundation produces DDI3/DDI4 CommandCode from SPSS, SAS, Stata, and R command scripts.
Whereas ComputationActions are used in statistical packages like SPSS, Stata, SAS and R to perform data management and data transformations, MetadataDrivenActions are used by ETL (Extract Transform Load) platforms along with ComputationActions.
What is new in DDI4: Process Decomposition

Whereas ComputationActions are used in statistical packages like SPSS, Stata, SAS and R to perform data management and data transformations, MetadataDrivenActions are used by ETL (Extract Transform Load) platforms along with ComputationActions.

In ETLs the user is presented with a menu of MetadataDrivenActions. In ETLs users enter into a dialog with the platform through which they customize the MetadataDrivenAction. The user writes no code. The dialog is saved as a MetadataDrivenAction.
In ETLs the user is presented with a menu of MetadataDrivenActions. In ETLs users enter into a dialog with the platform through which they customize the MetadataDrivenAction. The user writes no code. The dialog is saved as a MetadataDrivenAction.

Whereas ComputationActions are used in statistical packages like SPSS, Stata, SAS and R to perform data management and data transformations, MetadataDrivenActions are used by ETL (Extract Transform Load) platforms along with ComputationActions.
In a ConceptualInstrument two types of actions are supported – asking a Question and taking a Measurement.
What is new in DDI4: Process Decomposition

Question and Measurement are each associated with a RepresentedVariable and, through that variable, each has a ResponseDomain. That being said, a Question and a Measurement are each distinct specializations of the InstrumentComponent, making traversals of both “legacy” and new data sources possible.

In a ConceptualInstrument two types of Acts are supported – asking a Question and taking a Measurement.
DDI4 Prototype

Use Case Based Packages

Relation Type (WholePart, GeneralSpecific, ParentChild), Totality and Semantics by Level

Control Constructs WorkflowStepSequence

GLBPM GSBPM Domain-specific BPMs Prospective and Retrospective SOA and Integration Platform as a Service (iPaaS) Support

Questionnaires Medical Procedures Registry and Internet Query Sequences

Unit Data Arrangements Aggregates Information Models

https://www.dropbox.com/s/0ga88xs24pfxeo7/The%20DDI4%20Prototype.pptx?dl=0
Parking Lot

*The StatisticalClassificationView uses both DDI4 simple and structured collections to describe a ClassificationFamily that consists of two or more ClassificationSeries of StatisticalClassifications*
The StatisticalClassificationView (1 of 5)

- **StatisticalClassification** inherits from **CodeList**
- StatisticalClassification is either a SimpleCollection or a StructuredCollection
- As a StructuredCollection, StatisticalClassification is structured by a **ClassificationRelationStructure**
This ClassificationRelationStructure is at the core of all Statistical Classifications

- ClassificationRelationStructure, like other RelationStructures in DDI4, is a list of adjacency lists
- In this list of lists each vertex has a list of its adjacent vertices
- In the context of a StatisticalClassification vertices are Categories, and each Category has a list of adjacent Categories
This ClassificationRelationStructure is at the core of all StatisticalClassifications.

ClassificationRelationStructure, like other RelationStructures in DDI4, is a list of adjacency lists. In this list of lists each vertex has a list of its adjacent vertices. In the context of a StatisticalClassification vertices are Categories, and each Category has a list of adjacent Categories.
• The relationship between a Category and its adjacencies can be WholePart, ParentChild or GeneralSpecific
• In a ClassificationRelationStructure, each pair consists of a vertex and its adjacencies and corresponds to a Level
• From one Level to the next these relationships may change or be the same
The relationship between a Category and its adjacencies can be WholePart, ParentChild or GeneralSpecific.

In a ClassificationRelationStructure, each pair consists of a vertex and its adjacencies and corresponds to a Level. From one Level to the next these relationships may change or be the same.
The DDI4 Prototype 38

The StatisticalClassificationView (4 of 5)

- StatisticalClassifications form a `ClassificationSeries` when StatisticalClassifications are related to each other as versions or updates.
- A collection of ClassificationSeries in turn forms a `ClassificationFamily`.
  - Within a ClassificationFamily a collection of ClassificationSeries are loosely coupled around a common Concept or point of view.
By way of example, a family of industry classifications is composed of a list of two ClassificationSeries – NAICS and US-SIC.

Each ClassificationSeries is structured by a StatisticalClassification RelationStructure that takes the form of a version tree.
The StatisticalClassificationView (5 of 5)

• By way of example, a family of industry classifications is composed of a list of two ClassificationSeries – NAICS and U.S.-SIC.

• Each ClassificationSeries is structured by a StatisticalClassificationRelationStructure that takes the form of a version tree.
By way of example, a family of industry classifications is composed of a list of two ClassificationSeries – NAICS and US-SIC.

Each ClassificationSeries is structured by a StatisticalClassificationRelationStructure that takes the form of a version tree:
By way of example, a family of industry classifications is composed of a list of two ClassificationSeries – NAICS and US-SIC.

Each ClassificationSeries is structured by a StatisticalClassificationRelationStructure that takes the form of a version tree.
SPSS, SAS, Stata, R Command Script

Script Parser

Standard Data Transformation Language (JSON)

C² Metadata Project

DDI4 CommandCode ActivityDescription and RelatedMaterial