The DDI Data Management View: Prospective and Retrospective Workflow Description

Jay Greenfield, Ph.D.
Data Scientist
Data Documentation Initiative

Abstract: In DDI4 we have developed a Data Management View that describes the prospective and retrospective use of multiple data management platforms and architectures including (1) ESB (Enterprise Service Bus) and SOA (Service Oriented Architecture); (2) the use of PROCs/Commands in statistical packages like SAS, Stata and R; and now (3) the use of iPaaS (integration Platform as a Service) in public clouds, private clouds and apparatuses as practiced by various ETL (Extract, Transform and Load) platforms.

At one level the Data Management View consists of a data pipeline that traverses a series of business activities from business process models like the GSBPM (Generic Statistical Business Process Model) for the production of statistics and the GLBPM (Generic Longitudinal Business Process Model) for the description of longitudinal studies. At another level the Data Management View decomposes these business processes into a series of workflow steps. At both levels components exchange data.

Background

The ALPHA Network (Analyzing Longitudinal Population-based HIV/AIDS data in Africa) and its migration from Stata to Pentaho Data Integration has been an important use case for the DDI4 Data Management View. In DDI 3 perhaps the most formative data management use case was data exchange during data capture (with questionnaires) and between data capture and the data archive. DDI4 data management description aims to account for the ingestion and production of new data types (registry data, health data, big data, spell data, event data, etc.) and both legacy and new data management services that give shape to these data types in the course of the data lifecycle.

1. DataPipeline

With the Data Management View, the user is able to construct a DataPipeline of BusinessProcesses where each BusinessProcess contains either a simple collection or a structured collection of WorkflowSteps.

The DataPipeline itself is a simple collection of BusinessProcesses with the next one beginning after the preceding one ends.

Use the Data Management View and its DataPipeline to traverse a business process model once to describe the data lifecycle of a Study and many times to describe the data lifecycle of a StudySeries.
2. **BusinessProcess**

A **BusinessProcess** has an **AlgorithmOverview**, zero or more **Preconditions**, zero or more **Postconditions** and one or more **StandardModelUsed**.

Note that **Preconditions and Postconditions are LogicalRecords**. One **BusinessProcess** creates/updates LogicalRecords as **Postconditions**. These **Postconditions** may become the **Preconditions** of the next **BusinessProcess**.

![Diagram](image.png)

*Figure 1: The DataPipeline*

Note that support for business process models here is not limited to either the GSBPM or the GLBPM that the GSBPM has spawned.

The **Generic Statistical Business Process Model (GSBPM)** circumscribes a set of business processes that together describe a data lifecycle for building a single wave of national statistics. The **Generic Longitudinal Business Process Model (GLBPM)** specializes the GSBPM for longitudinal studies. Here each traversal of the GLBPM corresponds to the data lifecycle of a **Study** in a **StudySeries**.

Other business process models are more or less based on the GLBPM and the GSBPM. For example, a widely used demographic and epidemiological surveillance business process model specializes the GLBPM data lifecycle to create events, event histories and spells.
3. **WorkflowSteps**

The BusinessProcess decomposes into a WorkflowStepSequence. Alternatively, the BusinessProcess AlgorithmOverview can be used to outline a WorkflowStepSequence. This would be appropriate in situations where only a higher-level description was needed.

![Figure 3: Example AlgorithmOverview](image)

The WorkflowStepSequence may be either a Simple Collection or a Structured Collection of WorkflowSteps. In a simple collection of WorkflowSteps, successive pairs in the sequence participate in a before/after relationship. In a structured collection, however, relationships among the WorkflowSteps can be complicated: a sequence may have multiple starting points and multiple end points. Here before and after relationships can be indeterminate, depending on a platform and its technology stack.

The structured collection of WorkflowSteps is called a WorkflowStepRelationStructure. Like other structured collections in DDI4, a WorkflowStepRelationStructure is specified in a graph form as an unordered list of adjacency lists where for each vertex we specify an array of adjacent vertices:
Here each vertex and the array of its adjacencies are WorkflowSteps.

```xml
<WorkflowStepRelationStructure>
  <Agency>ALPHA_NETWORK</Agency>
  <Id>WorkflowStepRelationStructure:Individuals residences branch</Id>
  <Version>1</Version>
  <HasRelationSpecification>ImmediateDependency</HasRelationSpecification>
  <HasMemberRelation xmlns="" typeOfClass="WorkflowStep">
    <Source alias="TableInput:1">urn:ALPHA_NETWORK:Table input - Individuals</Source>
    <Target alias="SortRows:4">urn:ALPHA_NETWORK:Sort rows</Target>
  </HasMemberRelation>
  <HasMemberRelation xmlns="" typeOfClass="WorkflowStep">
    <Source alias="SortRows:4">urn:ALPHA_NETWORK:Sort rows</Source>
    <Target alias="MergeJoin:3">urn:ALPHA_NETWORK:Merge Join</Target>
  </HasMemberRelation>
  <HasMemberRelation xmlns="" typeOfClass="WorkflowStep">
    <Source alias="SortRows:10">urn:ALPHA_NETWORK:Sort rows indiv indiv residences</Source>
    <Target alias="MergeJoin:7">urn:ALPHA_NETWORK:Merge Join indiv indiv residences & Structure</Target>
  </HasMemberRelation>
  <HasMemberRelation xmlns="" typeOfClass="WorkflowStep">
    <Source alias="MergeJoin:7">urn:ALPHA_NETWORK:Merge Join indiv indiv residences & Structure</Source>
    <Target alias="MemoryGroupBy:11">urn:ALPHA_NETWORK:Memory Group by - indiv indiv residence & B Structure</Target>
  </HasMemberRelation>
  <HasMemberRelation xmlns="" typeOfClass="WorkflowStep">
    <Source alias="TableOutput:12">urn:ALPHA_NETWORK:Table output - indiv_res bs</Source>
    <Target alias="SortRows:5">urn:ALPHA_NETWORK:Sort rows 2</Target>
  </HasMemberRelation>
  <HasMemberRelation xmlns="" typeOfClass="WorkflowStep">
    <Source alias="SortRows:5">urn:ALPHA_NETWORK:Sort rows 2</Source>
    <Target alias="MergeJoin:3">urn:ALPHA_NETWORK:Merge Join</Target>
  </HasMemberRelation>
</HasMemberRelation>
```

Figure 4: The DDI4 RelationStructure

Figure 5: An Example WorkflowRelationStructure (fragment)
The Structured Collection in DDI4 is the successor to GSIM’s Node Set. Using graph representation, DDI4 RelationStructures, like the WorkflowStepRelationStructure, are easy to visualize and annotate using open source tools:

![Graph Rendering of Example DDI4 WorkflowRelationStructure](image)

**Figure 6: Graph Rendering of Example DDI4 WorkflowRelationStructure**

4. **Data Management View Use Cases**

- Create repeatable processes across a data network. More specifically, document and share the specifications for a demographic and epidemiological surveillance DataPipeline across surveillance sites
- Produce a Data Management Plan (DMP) in the form of a DataPipeline so other researchers are able to replicate a study’s results
- Document the actual data management in a study as a DataPipeline. This could underpin workflow tools for researchers
- Use a DataPipeline description as the input to tools that trace the lineage of data during the data lifecycle of a Study
- Use a DataPipeline description and the GraphML it spawns to create workflow diagrams
- Specialize the GLBPM to support the production of a dataset of geotagged tweets from the US where a wave corresponds to a day. Create a DataPipeline that describes in detail how this dataset is produced.
- Programmatically create a Data Management View for an Extract, Transform and Load (ETL) platform using the ETL’s authoring environment and the instructions that authors create as input
5. Target Audiences

- Researchers who are preparing a Data Management Plan (DMP)
- Data networks migrating from an Enterprise Service Bus (ESB) / Service Oriented Architecture (SOA) platform to a virtual (cloud-based) or actual integrated Platform as a Service (iPaaS) appliance (e.g. ETLs)
- Industry-specific or generic standard groups who wish to integrate fully developed information models with domain-specific business process models
- Search engines intent on exposing data lineage within a study

6. Included Classes

- AlgorithmOverview
- BusinessAlgorithm
- BusinessProcess
- ComputationAction
- Concept
- Coverage
- DataPipeline
- DataStore
- DesignOverview
- ExternalMaterial
- IfThenElse
- InstanceVariable
- InstanceVariableRelationStructure
- LogicalRecordRelationStructure
- Loop
- MetadataDrivenAction
- MethodologyOverview
- Parameter
- RepeatUntil
- RepeatWhile
- SpatialCoverage
- Split
- SplitJoin
- StructuredWorkflowSteps
- Study
- TemporalCoverage
- TopicalCoverage
- UnitDataRecord
- WorkflowMasterSequence
- WorkflowStepRelationStructure
- WorkflowStepSequence
7. Restricted Classes

The XML binding incorporates restrictions on relations between classes in the XML schema that corresponds to each view. Currently, there is not a comparable way of enforcing restrictions with RDF instances. Instead, RDF instance builders need to be mindful of the restrictions specified here. In the future, ShEx and/or SHACL descriptions may be provided to support RDF instance validation.

- In the DDI4 model a PhysicalRecordSegment maps to a LogicalRecord. This is not a relationship being supported in the Data Management View.

- In the DDI4 model instance variables participate in the so-called “variable cascade” where they enjoy direct or indirect relations with represented variables, conceptual variables and concepts. In the Data Management View, we slice and dice the DDI4 model in such a way that these direct and indirect relationships are severed.

- Study has been included in the view because a DataPipeline is specific to a Study. The relationships of Study, however, are not supported by this view with the exception of InstanceVariable, AlgorithmOverview, MethodologyOverview and DesignOverview.

- Many classes in this view directly or indirectly extend the abstract class WorkflowStep. These classes are Split, SplitJoin, IfThenElse, Loop, RepeatUntil, and RepeatWhile. They all restrict the use of WorkflowService in this View.

- DesignOverview has relationships to Precondition, and Goal that are not included in this view.

- InstanceVariable has relationships to many classes not included in this view. Information for many of these will need to be entered via other views. The restricted classes are: SubstantativeValueDomain, SentinelValueDomain, Concept, SubstantativeConceptualDomain, SentinelConceptualDomain, UnitType, Universe, Population, ConceptualVariable, RepresentedVariable, and Capture.

8. Next Steps

The Data Management View is part of the DDI4 Prototype which will be released in a May/June 2018 timeframe. Meanwhile it is being field tested with the ALPHA Network (Analyzing Longitudinal Population-based HIV/AIDS data in Africa). Here, with support from the Wellcome Trust and the Gates Foundation, a DDI4 Data Management View XML instance is being programmatically generated from the Pentaho Data Integration execution platform, Nesstar Publisher DDI 2.x metadata when available and a metadata infusion file prepared by the ALPHA team.