RESULTS-LEVEL METADATA: WHAT, HOW, AND WHY

Frank Dilorio, CodeCrafters, Inc.
Jeffrey Abolafia, Rho, Inc.
Some Background

CDISC: provide end-to-end standards

Already have “upstream” standards: CDASH, SDTM, ADaM
2015 release of the “other” end: Analysis Results Metadata (ARM)
This is an extension of the define-xml version 2 standard

CDISC now provides

End-to-end standards
Full traceability from data collection to results
SCOPE OF THIS PRESENTATION

Discuss history of ARM

Make the business case for its use

Outline implementation strategies

- Storage: Database
- Entry: Interface
- Coding: Creating the XML

Throughout: identify costs/benefits of different strategies
**History of ARM**

- **First machine-readable description of eSub**
  - Define.xml Version 1.0 (2005)
  - Focus on SDTM, but could be used for ADaM

- **ADaM Version 2.0 (2006)**
  - Introduced concept of ARM
  - No formal mechanism to submit ARM

- **Define.xml Version 2 released (2013)**

- **2015 Define XML Version 2 extension includes ODM schema for Results**
ARM RELEASE PACKAGE

Sounds good ... but where to start?
• As always, best place is the CDISC web site

www.cdisc.org/adam
• ZIP file with schema, specification document, and ADaM define.xml examples
Why Bother with ARM? It’s not required

It must be costly to modify databases, interfaces and software to produce ARM…

🔍 Look at benefits:

✔ The documentation contained in ARM aids traceability
✔ This in turn can hasten time to regulatory approval
✔ Same metadata can be used throughout the TFL life cycle as with datasets/variable metadata
  • For creation of TFLs (header/footer, subsetting, etc.)
  • As part of project management (discussed later)
IMPLEMENTATION

ARM “tells the story” of a TFL; how do we collect, store, and present the pieces of the story?

Next sections describe our ARM experience with:

- Database design
- User interface and metadata entry
- Creation of ARM-compliant define.xml

First, though, let’s read ahead and see the finished story ...
**TRANSFORMED ARM (FROM CDISC RELEASE PACKAGE)**

Style sheet transforms define.xml into HTML that presents each display in a readable, highly navigable/hyperlinked table.

### Table 14-3.01 Primary Endpoint Analysis: ADAS-Cog - Summary at Week 24 - LOCF (Efficacy Population)

- Dose response analysis for ADAS-Cog changes from baseline
- Pairwise comparisons to placebo for ADAS-Cog changes from baseline

### Table 14-5.02 Incidence of Treatment Emergent Serious Adverse Events by Treatment Group

- Incidence of Treatment Emergent Serious Adverse Events by Treatment Group

---

### Analysis Results Metadata (Detail) for Study CDISC-Sample

<table>
<thead>
<tr>
<th>Display</th>
<th>Table 14-3.01 Primary Endpoint Analysis: ADAS-Cog - Summary at Week 24 - LOCF (Efficacy Population)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis Result</td>
<td>Dose response analysis for ADAS-Cog changes from baseline</td>
</tr>
<tr>
<td>Analysis Parameter (s)</td>
<td>PARAMCD = &quot;ACTOT&quot; (Adas-Cog(11) Subscore)</td>
</tr>
<tr>
<td>Analysis Variable (s)</td>
<td>CHG (Change from Baseline)</td>
</tr>
</tbody>
</table>
TRANSFORMED ARM (FROM CDISC RELEASE PACKAGE)

Style sheet transforms define.xml into HTML that presents each display in a readable, highly navigable/hyperlinked table.
TELLING THE STORY, PART 1: THE DATABASE

“Chatty” story means lots of metadata

Database design needs to be carefully thought out

CDISC specification describes naming, content, status and cardinality of fields

Most of these are summarized in the next slide ...
## ARM Fields

<table>
<thead>
<tr>
<th>Level</th>
<th>Item</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>Unique display name-number</td>
<td>Yes</td>
</tr>
<tr>
<td>Result</td>
<td>Title of the Display</td>
<td>Yes</td>
</tr>
<tr>
<td>Result</td>
<td>Analysis reason</td>
<td>Yes</td>
</tr>
<tr>
<td>Result</td>
<td>Analysis purpose</td>
<td>Yes</td>
</tr>
<tr>
<td>Result</td>
<td>Description</td>
<td>Yes</td>
</tr>
<tr>
<td>Result</td>
<td>Reference to external documentation</td>
<td>No</td>
</tr>
<tr>
<td>Result</td>
<td>Description of how to join multiple datasets used</td>
<td>Conditional</td>
</tr>
<tr>
<td>Result</td>
<td>Documentation (in-line text)</td>
<td>No</td>
</tr>
<tr>
<td>Result</td>
<td>Documentation (reference to external document)</td>
<td>No</td>
</tr>
<tr>
<td>Result</td>
<td>Programming code</td>
<td>No</td>
</tr>
<tr>
<td>Result</td>
<td>Software reference (e.g. “SAS Version 9.3”)</td>
<td>No</td>
</tr>
<tr>
<td>Result</td>
<td>Link to the program that created the Result</td>
<td>No</td>
</tr>
<tr>
<td>Result-dataset</td>
<td>Name of dataset used for the Result</td>
<td>Yes</td>
</tr>
<tr>
<td>Result-dataset</td>
<td>Dataset filter (WHERE clause) for PARAMCD-based Result</td>
<td>Conditional</td>
</tr>
<tr>
<td>Result-dataset</td>
<td>Variables used to create the analysis result</td>
<td>Conditional</td>
</tr>
</tbody>
</table>
FIELDS AND REPETITION

• Note that just as ARM is not required for every display, there are also optional and conditional fields.

• “Level” implies hierarchy (cardinality is in bold):

  Display description, documentation [1]
  Result [1 or more]
  Description [1]
  Datasets and variables [1 or more]
  Documentation [1]
  Programming code [1]

The database must be able to accommodate multiple results, datasets per display.
Level of indentation corresponds with level of hierarchy:

```xml
<arm:ResultDisplay OID="RD.Table_14-5.02" Name="Table 14-5.02">
  <Description>
    <TranslatedText>Incidence of Serious AEs</TranslatedText>
  </Description>
  <def:DocumentRef leafID="LF.Table-14-5.02">
    <def:PDFPageRef PageRefs="3" Type="PhysicalRef"/>
  </def:DocumentRef>
  <arm:AnalysisResult OID="AR.Table_14-5.02.R.1">
    AnalysisReason="SPECIFIED IN SAP"
    AnalysisPurpose="PRIMARY OUTCOME MEASURE">
      <Description>
        <TranslatedText>Trt-Emergent Serious AEs</TranslatedText>
      </Description>
      <arm:AnalysisDatasets def:CommentOID="COM.JOIN-ADSL-ADAE">
        <arm:AnalysisDataset ItemGroupOID="IG.ADAE">
          <def:WhereClauseRef WhereClauseOID="WC.Table_14-5.02.R.1.ADAE"/>
          <arm:AnalysisVariable ItemOID="IT.ADAE.AEBODSYS"/>
        </arm:AnalysisDataset>
      </arm:AnalysisDatasets>
```
HIERARCHY ILLUSTRATED USING XMLPad
**TELLING THE STORY, PART 2: POPULATING ARM**

**2 Approaches**

**Interface: extend capabilities of existing system**
- Can’t just add new fields to existing “flat” structure. Have to add screens, links for metadata that repeats (results, datasets)

**Programmatic: populate XLS/other, and load into Oracle**
- Makes repetition of rows (e.g., results) easier than using the interface

Considering end-user needs, it’s best to have both options available!
POPULATING ARM: SAMPLE INTERFACE

Look and feel for ARM resembles what we currently use for entry of dataset/variable/value metadata
## Populating ARM: Sample Interface

<table>
<thead>
<tr>
<th>Name</th>
<th>Links</th>
<th>Status</th>
<th>Title</th>
<th>Population</th>
<th>Group</th>
<th>Number</th>
<th>Validation Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAD_DM_01 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Demographic and Baseline Characteristics</td>
<td>Intent-to-Treat</td>
<td>Demographics</td>
<td>14.1.3</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TAD_DM_02 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Demographic and Baseline Characteristics</td>
<td>Clinically Evaluable</td>
<td>Demographics</td>
<td>14.1.4</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TAD_DM_03 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Demographic and Baseline Characteristics</td>
<td>Enrolled Safety</td>
<td>Demographics</td>
<td>14.1.5</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TBI_EX_01 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Exposure to Final Dose by Final Dose Group</td>
<td>Enrolled Safety</td>
<td>Exposure and Compliance</td>
<td>14.3.5.5</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TBI_EX_02 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Exposure to Final Dose by Final Dose Group</td>
<td>Randomized Safety</td>
<td>Exposure and Compliance</td>
<td>14.3.5.6</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TAG_EX_01 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Exposure to Treatment</td>
<td>Enrolled Safety</td>
<td>Exposure and Compliance</td>
<td>14.3.5.1</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TAG_EX_02 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Exposure to Treatment</td>
<td>Randomized Safety</td>
<td>Exposure and Compliance</td>
<td>14.3.5.2</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TAH_FD_01 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Final Dose by Subgroups</td>
<td>Enrolled Safety</td>
<td>Exposure and Compliance</td>
<td>14.3.5.3</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TAH_FD_02 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Final Dose by Subgroups</td>
<td>Randomized Safety</td>
<td>Exposure and Compliance</td>
<td>14.3.5.4</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TAX_AEO_01 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Overview of Adverse Events During Double-Blind Phase</td>
<td>Randomized Safety</td>
<td>Adverse Events</td>
<td>14.3.1.3</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TAV_WO_03 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Overview of Adverse Events During Entire Study</td>
<td>Enrolled Safety</td>
<td>Adverse Events</td>
<td>14.3.1.4</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TAV_WO_01 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Overview of Adverse Events During Open-Label Phase</td>
<td>Enrolled Safety</td>
<td>Adverse Events</td>
<td>14.3.1.1</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TAV_WO_02 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Overview of Adverse Events During Open-Label Phase</td>
<td>Randomized Safety</td>
<td>Adverse Events</td>
<td>14.3.1.2</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TBE_VS_02 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Potentially Clinically Significant (PCS) Vital Sign Values During Double-Blind Phase</td>
<td>Randomized Safety</td>
<td>Vital Signs</td>
<td>14.3.5.10</td>
<td>Double Programming</td>
</tr>
<tr>
<td>TBE_VS_01 [Tasks]</td>
<td></td>
<td>Started</td>
<td>Potentially Clinically Significant (PCS) Vital Sign Values During Entire Study</td>
<td>Enrolled Safety</td>
<td>Vital Signs</td>
<td>14.3.5.9</td>
<td>Double Programming</td>
</tr>
</tbody>
</table>
TELLING THE STORY, PART 3: CREATING THE XML

Results XML is defined by an ODM schema extension (in CDISC release package)

Transformation into HTML is handled by style sheet/XSL (also in release package) that is ARM aware

Building Results XML is no different than building earlier versions of define.xml
Regardless of software used to create the XML, keep in mind:

- **OID (Object IDentifier) formation is critical**
  - `define.xsl` creates links from results section back to datasets & variables. Therefore, naming has to be consistent.

- **Exploit repetition**
  - Results XML has elements (e.g., Description, WhereClause) found elsewhere in define.xml. Turn creation of these elements into a macro/function.
OID FORMATION

Highlighted OIDs are part of non-Results XML. Naming must be identical here in Results in order for style sheet to create link successfully.

<arm:ResultDisplay OID="RD.Table_14-5.02" Name="Table 14-5.02">  
  <Description>  
    <TranslatedText>Incidence of Serious AEs</TranslatedText>  
  </Description>  
  <def:DocumentRef leafID="LF.Table-14-5.02">  
    <def:PDFPageRef PageRefs="3" Type="PhysicalRef"/>  
  </def:DocumentRef>  
<arm:AnalysisResult OID="AR.Table_14-5.02.R.1"  
  AnalysisReason="SPECIFIED IN SAP"  
  AnalysisPurpose="PRIMARY OUTCOME MEASURE">  
  <Description>  
    <TranslatedText>Trt-Emergent Serious AEs</TranslatedText>  
  </Description>  
  <arm:AnalysisDatasets def:CommentOID="COM.Join-ADSL-ADAE">  
    <arm:AnalysisDataset ItemGroupOID="IG.ADAE">  
      <def:WhereClauseRef WhereClauseOID="WC.Table_14-5.02.R.1.ADAE"/>  
      <arm:AnalysisVariable ItemOID="IT.ADAE.AEBODSYS"/>  
    </arm:AnalysisDataset>  
  </arm:AnalysisDatasets>
Making XML Viewable: The Style Sheet

- XML is plain text and virtually unreadable.
- XSL (style sheets) transforms XML into HTML when the XML is opened in Windows Explorer or similar.

Keys points:

- XSL is conceptually different than other languages (so, generally, a badly shaped learning curve)
- Use the CDISC style sheet (in release package) unless it doesn’t fully meet your needs.
Is the XML Correct? Validation

• Two levels of validation:
  ▪ Syntax: the easy one (errors displayed in browser if elements are not closed, attributes aren’t quoted correctly, etc.)
  ▪ Content ("semantic"): the harder one. Validation answers the question “is the content of the XML consistent with the ODM schema?”
    • Are elements in the correct order?
    • Are required elements and attributes present?
    • Do attributes have expected values?

• We have tools for this, right ...?
OpenCDISC Validation

OpenCDISC has become the *de facto* standard for dataset and define.xml validation

Current (2015Q3) status of define.xml results validation:
- Not in Community (free, open source) version
- Will be in Enterprise (fee-based) version

Community version users will need to write their own checks. These can be created within the OpenCDISC validation framework.
OTHER USES OF ARM

ARM is not just for “downstream”, eSub use

SAS macros or similar tools can use the metadata earlier in the study life cycle:

• Extract metadata to facilitate TFL production
• Create TFL template programs
• With additional fields in the database, can create macro variables containing TFL footnotes and other code fragments.

Management/tracking
Final Thoughts

Good for regulatory reviewers and sponsors

Can speed time to drug approval.

Creation of database, means of ARM entry, and XML is not trivial.

Thoughtful planning and design, especially in concert with existing tools, can facilitate addition of this valuable metadata.

Building the infrastructure requires an up front investment but worth the investment!!
Thank you!

• Your comments and questions are appreciated and valued

• Contact the authors
  - Frank Dilorio: frank_dilorio@rhoworld.com
  - Jeff Abolafia: jeff_abolafia@rhoworld.com