IASSIST & CARTO 2018

Once Upon a Data Point: Sustaining our Data Storytellers

C²METADATA

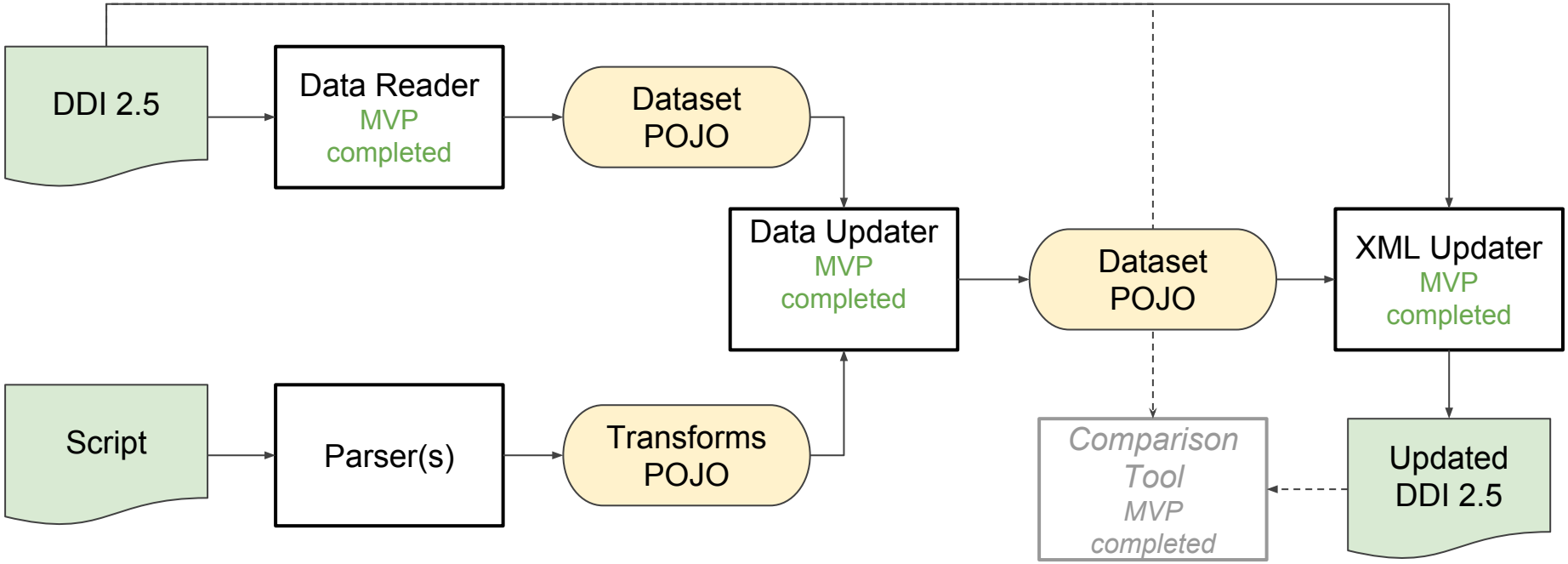
Continuous Capture of Metadata

Automating the Capture of Data Transformations from Scripts for Statistical Packages

metadata technology NORTH AMERICA

Jack Gager, Pascal Heus, Carson Hunter, Et al.
mtna@mtna.us

System Components



How it works?

Reader

- Takes DDI XML and turns it into our own dataset model POJOs (Plain Old Java Object)
  - POJOs = Dataset, Variable, Classification, Code, ...

Dataset Updater

- Takes SDTL and creates a Program object based on COGS model and Transforms (our own model) to apply changes one at a time to the dataset
- Uses a set of simple interface representing all transform use cases
  - SDTL commands implements relevant interfaces

XML Updater

- Takes dataset and updates the xml, producing a final xml file with the transformations listed as derivations

Comparison Tool

- Internal utility to analyze dataset differences (for documentation / validation purposes)

Supported Commands (MVP / planned)

Assignment	COMPUTE	Implemented	generate replace egen
Conditional assignment	IF	todo	replace … if
Recode	RECODE	Implemented	recode
Select cases	SELECT IF	todo	drop if keep if
Select variables	DELETE VARIABLES	implemented	drop keep
Define missing values	MISSING VALUES	Implemented	
Label variables and values	VARIABLE LABELS VALUE LABELS	Implemented	label
Format	PRINT FORMATS WRITE FORMATS	Implemented	format
Rename variables	RENAME	Implemented	rename
Re-order variables	SORT VARIABLES	todo	order
File merge (SQL Append, Join, Aggregate)	ADD FILES MATCH FILES AGGREGATE	ADD FILES implemented	append merge egen collapse

Dataset Updater

All Transformation can be expressed by a simple set of variable or file level operations

SdtlWrapper base interface that carries original transform → extended by general instructional command interfaces:

- LoadsDataset
- JoinsDatasets
- DeletesVariable
- CreatesVariables
- ReordersDataset
- SelectsVariable
- UpdatesVariables
- UpdatesClassification
- SavesDataset

(order is important)

Wrapper classes that wrap a SDTL Transform Base command and implement one or more of our interfaces. Most of the work of parsing variables is done in-class aside from ranges, which will be expanded in the updater with the dataset in scope.

- Compute
- Delete
- Load
- Merge
- Recode
- Rename
- Save
- SetMissingValues
- SetValueLabels
- SetVariableLabel
- UnknownCommand

Updated DDI

- Holds both the source, target (and intermediate) datasets
  - one fileDscr / dataDscr per dataset
- target vars stripped of summary / descriptive statistics, ranges, etc.
  - new data statistics can be computed externally and merged
  - considering option to carrying over some descriptive elements
- var/derivation holds transformation information
  - derivation/@var point to source variables
- /derivation/drvCmd holds individual commands applied to the var
  - scripts syntax, SDTL JSON, plain English
  - absence of drvCmd means var carried over
  - DDI-C caveat: no @var at drvCmd level

Recode Example

<ddi>

```
<var ID="V5" name=" EDUC2" files="F2">
  <labl>EDUCATION</labl>
  <catgry><catValu> 0</catValu><labl> None</labl></catgry>
  <catgry><catValu> 1</catValu><labl> Grade School</labl></catgry>
  <catgry><catValu> 2</catValu><labl> High School</labl></catgry>
  <catgry><catValu> 3</catValu><labl> College</labl></catgry>
  <derivation var="V2">
    <drvcmd source="producer" syntax="">recode V520131 (0=0) (1,2=1) (3 thru 6=2) (7,8=3) into EDUC2.</drvcmd>
    <drvcmd source="producer" syntax="">value labels EDUC2 1 'Grade School' 2
'None'.</drvcmd>
  </derivation>
</var>

<var ID="V4" name=" EDUC1" dcml="0" intrvl="discrete" files="F2">
  <labl>EDUCATION</labl>
  <catgry><catValu>0</catValu><labl>NONE</labl></catgry>
  <catgry><catValu>1</catValu><labl>SOME GRADE SCHOOL</labl></catgry>
  <catgry><catValu>2</catValu><labl>COMPLETED GRADE SCHOOL</labl></catgry>
  <catgry><catValu>3</catValu><labl>SOME HIGH SCHOOL</labl></catgry>
  <catgry><catValu>4</catValu><labl>COMPLETED HIGH SCHOOL</labl></catgry>
  <catgry><catValu>5</catValu><labl>INCOMPLETE HIGH SCHOOL PLUS OTHER NON-</
  <catgry><catValu>6</catValu><labl>COMPLETED HIGH SCHOOL PLUS OTHER NON-C</
  <catgry><catValu>7</catValu><labl>SOME COLLEGE</labl></catgry>
  <catgry><catValu>8</catValu><labl>COMPLETED COLLEGE (HAS A DEGREE)</labl></catgry>
  <catgry missing="Y"><catValu>9</catValu><labl>NA, OR NO PRE-ELECTION INTERVIEW
(CODE</labl></catgry>
  <derivation var="V2">
    <drvcmd source="producer" syntax="">rename variables (V520131=EDUC1).</drvcmd>
  </derivation>
  <varFormat type="numeric" schema="other"/>
</var>
```

SPSS

```
*recode existing (categorical) variable into new
variable.
*assign value labels to new variable.
*rename existing (old) variable.

get file='da07213_inputForRecode'.
recode V520131 (0=0) (1,2=1) (3 thru 6=2) (7,8=3) into
EDUC2.
value labels EDUC2 1 'Grade School' 2 'High School' 3
'College' 0 'None'.
rename variables (V520131=EDUC1).
save outfile='da07213_Recode_ValueLabels'.
EXECUTE.
```

Recode Example

```
<var ID="V5" name="EDUC2" files="F2">
  <labl>EDUCATION</labl>
  <catgry><catValu>0</catValu><labl>None</labl></catgry>
  <catgry><catValu>1</catValu><labl>Grade School</labl></catgry>
  <catgry><catValu>2</catValu><labl>High School</labl></catgry>
  <catgry><catValu>3</catValu><labl>College</labl></catgry>
  <derivation var="V2">
```

```
recode V520131 (0=0) (1,2=1) (3 thru 6=2) (7,8=3) into
EDUC2.
value labels EDUC2 1 'Grade School' 2 'High School' 3
'College' 0 'None'.
```

Some things to consider…

- Make sure DDI is the right version (and how to update, + status on what versions we will accept)
- Make sure script has the right file name (especially if it was just transformed)
- Single file uploads for now

```
<fileDscr ID="F1" URI="Temp#17.~esstar?Index=0&amp;Name=da
   <fileTxt>
      <fileName>da07213_inputForRecode.NSDstat</fileName>
      <dimensns>
```

```
use "da07213_inputForRecode" , clear
recode V520131 (0=0) (1 2=1) (3/6=2) (7 8=3), gener(EDUC2)
lab def Edlab 1 "Grade School" 2 "High School" 3 "College" 0 "Non
lab val EDUC2 Edlab
rename V520131 EDUC1
saveold "da07213_Recode_ValueLabels.dta" , version(12) replace
```

Comparison Tool

- Early in this project, we wrote a tool to report on transformations in datasets.
- By running the source and final DDI through the same tool, we can get a JSON report of the changes and make sure they align with what we were expecting from the script.
- Comparison Tool
  - Website has documentation and instructions
- DEMO

Using the Comparison Tool

Get a client key:

GET http://localhost:8080/DatasetComparisonService/clientKey

```
1 "22691394-722c-476a-aa18-716e894409ef"
```

Upload documents:

POST http://localhost:8080/DatasetComparisonService/upload/ddi2?clientKey=22691394-722c-476a-aa18-716e894409ef x2

```
1 {
2     "Temp3": "b6ce758e-434b-44bb-8f93-69ea1521b0c0",
3     "SAMPLE_XU_OUTPUT": "14fcf59e-230f-4035-93e4-382417cbf3de"
4 }
```

Compare documents:

GET http://localhost:8080/DatasetComparisonService/compare/datasets?sourceKey=___&targetKey=__

Comparison Example

SPSS

```
*recode existing (categorical) variable into new variable.
*assign value labels to new variable.
*rename existing (old) variable.

get file='da07213_inputForRecode'.
recode V520131 (0=0) (1,2=1) (3 thru 6=2) (7,8=3) into
EDUC2.
value labels EDUC2 1 'Grade School' 2 'High School' 3
'College' 0 'None'.
rename variables (V520131=EDUC1).
save outfile='da07213_Recode_ValueLabels'.
EXECUTE.
```


```
{
  "service": "DDI2.5 Dataset Comparison",
  "request": {
    "timestamp": "Tuesday, May 22, 2018 2:02:25 PM EDT" },
  "results": {
    "dataSetComparisonResults": {
      "source": "Temp17.F1",
      "target": "SAMPLE_XU_OUTPUT.F2",
      "attributes": [ {
          "name": "dataSetNames",
          "sourceValue": "Temp#17",
          "targetValue": "SAMPLE_XU_OUTPUT",
          "match": false
        }, {
          "name": "variableCount",
          "sourceValue": 2,
          "targetValue": 3,
          "match": false,
          "difference": "GREATER_THAN"
        }, {
          "name": "classificationCount",
          "sourceValue": 1,
          "targetValue": 1,
          "match": true
        }... ],
      "statistics": {
        "variableStatistics": {
          "newVariableCount": 2,
          "droppedVariableCount": 1,
          "updatedVariableCount": 0,
          "matchedVariableCount": 1
        },
        "classificationStatistics": {
          "newClassificationCount": 0,
          "droppedClassificationCount": 0,
          "updatedClassificationCount": 0,
          "matchedClassificationCount": 1
        }
        "classificationStatistics": {
          "newClassificationCount": 0,
          "droppedClassificationCount": 0,
          "updatedClassificationCount": 0,
          "matchedClassificationCount": 1
        }
      }
    },
    "variableComparisonResults": {
      "variables": {
        "newVariables": [
          "EDUC1",
          "EDUC2"
        ],
        "droppedVariables": [
          "V520131"
        ]
      }
    }
  },
  "config": {...          }

    "classificationComparison": {
      "compared": true,
      "codeCount": true,
      "label": true
    }
  }
}
```

Comparison Example

```
recode V520131 (0=0) (1,2=1) (3 thru 6=2) (7,8=3) into
EDUC2.
value labels EDUC2 1 'Grade School' 2 'High School' 3
'College' 0 'None'.
rename variables (V520131=EDUC1).
```


```
[... ],
"statistics": {
  "variableStatistics": {
    "newVariableCount": 2,
    "droppedVariableCount": 1,
    "updatedVariableCount": 0,
    "matchedVariableCount": 1
  },
```

```
},
"variableComparisonResults": {
  "variables": {
    "newVariables": [
      "EDUC1",
      "EDUC2"
    ],
    "droppedVariables": [
      "V520131"
    ]
  }
```

What's next

- Support additional commands (based on parsers implementation)
- Auto upgrade of DDI < 2.5 + Nesstar DDI cleansing (cr/lf/spaces)
- Adjust command wrappers based on SDTL model changes
  - note: interfaces remain the same
- Test with more complex scripts (longer, intermediate files, etc.)
- Collect feedback from early adopters / users
- Support for other standards (e.g. EML) and syntaxes (parsers)
- We've learned a lot…
  - …and will likely continue to do so and improve…