DDI 3: EXTRACTING METADATA FROM THE DATA ANALYSIS WORKFLOW

By Larry Hoyle and Joachim Wackerow
with Oliver Hopt

2/2/2010

DDI Working Paper Series -- Use Cases, No. 4

This paper is part of a series that focuses on DDI usage and how the metadata specification should be applied in a variety of settings by a variety of organizations and individuals. Support for this working paper series was provided by the authors' home institutions; by GESIS - Leibniz Institute for the Social Sciences; by Schloss Dagstuhl - Leibniz Center for Informatics; and by the DDI Alliance.

DDI 3: EXTRACTING METADATA FROM THE DATA ANALYSIS WORKFLOW

BY LARRY HOYLE AND JOACHIM WACKEROW WITH OLIVER HOPT

ABSTRACT

In many instances the only source of certain metadata may be in a file saved from some data analysis program. This is an exploration of what metadata can be harvested from several commonly used programs, and therefore by deduction what else is not available from these programs. These metadata elements are mapped into the appropriate DDI 3 structure.

BACKGROUND

A few tools have been developed to extract metadata from SPSS, Stata, and SAS datasets (see http://tools.ddialliance.org/?lvl1=library). Each of these data analysis packages allows for a different set of metadata in their data files. For each of these packages the focus of the metadata they contain is primarily for the use of the data and not for other purposes such as resource discovery, citation, and context.

In some cases there are differences in the programs' underlying metadata models. In a few instances these differences may make it difficult to represent the complete set of metadata available in DDI[1]. An explicit enumeration of what metadata are available would be useful for future refinement of extraction programs and possible evolution of the DDI standard.

Some of these programs allow for representation of complex relationships between multiple tables. For the most part those features have been not been thoroughly explored in this analysis.

USE CASE / REQUIREMENTS

Software Packages Explored

The packages for which some tools have already been developed – SPSS, Stata, and SAS – were explored along with R, Microsoft Excel and Access, JMP, and the Triple-S XML standard. Comma-separated variable files (CSV), being a common export format, were included as well. StatTransfer, which can move data (and metadata) among any of these packages, was also evaluated.

[1] For this paper, "DDI" refers to the current version 3.1 of DDI.

Metadata may be extracted from files in the native format of these packages through several approaches. In several cases metadata may be extracted by a script native to the package. An external program such as DExT, or StatTransfer may also be able to extract both data and metadata.

RESULTS

CSV – Comma Separated Variable File

Metadata	Where to Find it
VARIABLES	
name	corresponding element in header row
basic data type	default is numeric, quoted indicates text, assumed for date
position	relative order in record, separated by a delimiter like comma, colon, or tab
VALUES	
missing	empty field expressed by consecutive commas

Comments – (CSV)

CSV files carry a very minimal set of metadata. In their minimal form the only explicit metadatum is the delineation of columns. Optionally, quotation marks may indicate the data type of text elements and a header line may indicate a column name.

Issues / Restrictions – (CSV)

In common usage, metadata are sometimes assumed when importing a CSV file. Strings like "12/21/2112" are interpreted as dates, and columns with anything other than numbers are assumed to be all character.

It might be possible to include other metadata at the beginning of CSV files. Some import programs are able to skip initial lines in the file, although this may also interfere with the ability to pull column names from the first line of the file. On the other hand, a companion DDI file could be created by any tools that could add structured metadata to a CSV file.

Microsoft Excel

Metadata	Where to Find it
Dataset	
name	may have range or sheet name
VARIABLES	
name	top cell in column of range or sheet
basic data type	text, number, date, formula
display format	format cells
position	order within column
decimal positions	format cells
specific type	format cells
VALUES	
missing	empty cell
notes	notes attached to cells
INTEGRITY CONSTRAINTS	
range restrictions	data validation
list restrictions	data validation
restriction by expression	data validation
Scripting Language	Visual Basic
comments	begin line with single quote

Comments – (Excel)

Spreadsheets, the original "killer app" for personal computers, are still a common tool for entry and some simple analysis of data. Metadata are present primarily as cell attributes, although various other metadata could be included in multiple sheets.

Cell attributes are accessible either programmatically through the API or directly from the XML .xlsx file.

Issues / Restrictions – (Excel)

A variety of metadata elements beyond those listed above are probably embedded in Excel spreadsheets, but not in any consistent way. It would be possible to set standards for the use of multiple sheets to represent different metadata elements, but, given the diverse and widespread use of Excel, probably impossible to have a large community actually use those standards. Within an organization or small community, though, it would be possible to devise machine-actionable representation of DDI metadata within a spreadsheet. For the wider community, extracting those metadata by hand, such as by cut and paste, will probably be the common solution.

Structured comments in VBA modules are a possibility for embedding metadata, but might not be very transparent to a typical spreadsheet user. Data structured within the VBA modules would be another, if even less transparent, option.

R

Metadata	**Where to Find it**
Dataset	
name	frame name
VARIABLES	
name	column name
basic data type	character, logical, number
display format	formatC, prettyNum
position	order in column
decimal positions	formatC digits
specific type	typeof(object)
measurement level	factors are nominal or ordinal if ordered
user defined attributes	new R classes can be defined
VALUES	
missing	NA special value
Scripting Language	R language
comments	begin line with #

Comments – (R)

We have restricted the evaluation of R in the preceding table to the basic R facility. One of the strengths of R is its large number of additional packages. There are some possibilities here. One example is the package spssDDI (http://cran.r-project.org/web/packages/spssDDI/index.html) which reads SPSS system files, structures the metadata into R lists, and produces DDI 3.0 documents. R lists are capable of representing complex metadata. The spssDDI readSpssSav function returns a list which contains the following metadata (see http://cran.r-project.org/web/packages/spssDDI/spssDDI.pdf):

A header: (RecordType, ProductName, LayoutCode, CaseSize, Compress, WeightIndex, NumCases, Bias, CreationDate, CreationTime, FileLabel)

Variables metadata: (Number missings, Print fmt, Write fmt, Varname, Varlabel, Missing values, Typecode, Format Type, (String, Numeric, Date, Time, Other))

And more: (Value labels, Sysmis, highest, lowest, Major, minor, revision, floating, endianness (big-endian or little-endian), character, Short and long varnames, Variable sets, Very long variables, Document Notes, Trend, Display, Strings' value labels, Encoding, Other)

This approach could be made more general and could include much of the DDI structure.

Issues / Restrictions – (R)

Given the extensibility of R it is possible to envision a DDI Core package which would define a data structure to hold a set of DDI-compatible metadata. R is capable of storing such a structure along with any associated data in its .RData format. A full-blown DDI editor could be developed as an R package, much like the R Commander graphical user interface to R.

This little snippet of code shows the construction of a list "MyList" which contains an empty list of Authors and a blank text value for "Universe". Values are then assigned to Authors and Universe and then values for the whole structure are listed.

```
> MyList <- list(Authors=list(" "), Universe=" ")
> MyList$Authors=list("Joe Schmoe", "Arti Ficial")
> MyList$Universe="Persons Over 65 Years of Age"
>
> MyList
$Authors
$Authors[[1]]
[1] "Joe Schmoe"

$Authors[[2]]
[1] "Arti Ficial"


$Universe
[1] "Persons Over 65 Years of Age"
```

Whether such a facility, particularly for longer and perhaps structured elements, would be attractive to users of R is an open question.

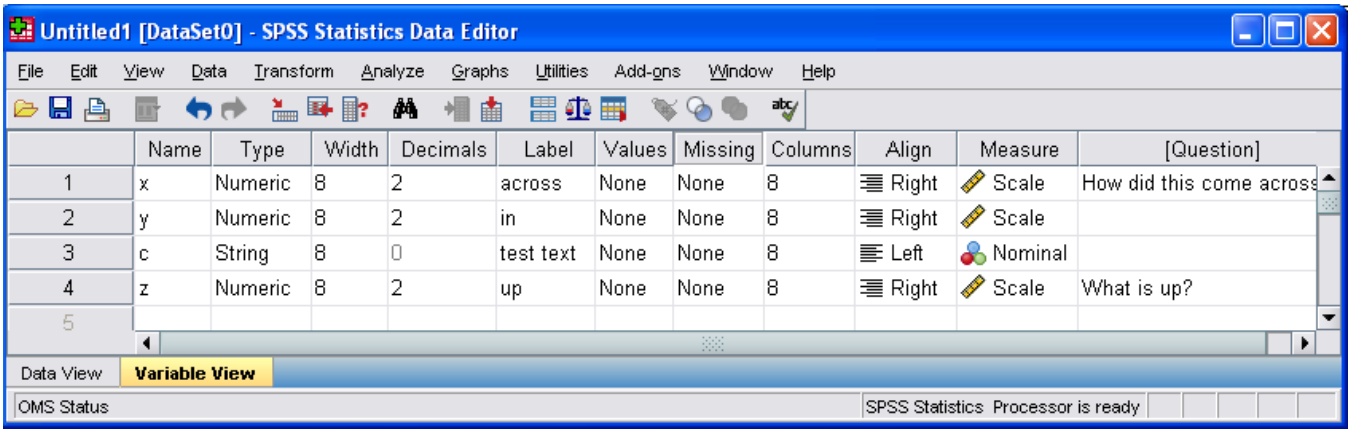
Some form of structured comments could also be used with R. These could be captured from interactive sessions in the .RHistory file.

SPSS

Metadata	Where to Find it
Dataset	
name	save file name
label	dataset label
VARIABLES	
name	variable name
basic data type	type
display format	format
position	position within dataset
label	variable label
decimal positions	format
specific type	format
weight	weight variable
measurement level	measure
user defined attributes	Variable view
VALUES	
missing	missing values
values as missing	can list data values to be treated as missing
ranges missing	a range can be included in the missing list
values can be labeled	value labels are assigned to the variable
Scripting Language	
comment	multi line, begins with * ends with .

Comments – (SPSS)

SPSS offers the capability for user defined attributes for variables with the option to display those attributes in the Variable View. In the figure below an attribute "Question" has been defined for the variables "x" and "z".



This facility would offer the capability to enter some additional DDI compatible metadata from within the SPSS user interface. A standard SPSS script could create a set of DDI core attributes to be added to the dataset. In the example below a script adds the attributes "Universe" and "Question". Display of the attributes in the Variable View can then be enabled in the user interface (View…Customize Variable View).

```
VARIABLE ATTRIBUTE
    VARIABLES=ALL
    ATTRIBUTE=Universe(' ') Question(" ") .
```

Issues / Restrictions – (SPSS)

There are multiple options for reading metadata from SPSS system files (see http://tools.ddialliance.org/?lvl1=library). These include:

- DExT
- Using the API to control SPSS
    - Python API Custom SPSS command (DDIWrite - Wackerow)
- Output information to SPSS Output Management System (OMS) XML format and then convert to DDI or an intermediate
    - STATSPROGS2DDI (Wackerow)
- spssDDI package in R
- SPSS Export files
    - no currently working tools

As with the other packages there would also be the possibility of using structured comments to capture metadata. SPSS allows scripting in its native language "SPSS Syntax", Visual Basic, and Python.

Stata

Metadata	Where to Find it
Dataset	
name	dataset name
label	dataset label
user defined attributes	characteristics
VARIABLES	
name	variable name
basic data type	type
display format	format (separate from label)
position	order within list of variables
label	variable label
decimal positions	format
specific type	type
user defined attributes	characteristics
notes	multiple notes per variable (numbered)
VALUES	
missing	multiple, see below
multiple distinct system missing	multiple system missing values - .a .b …. .z the values .a to .z may be labeled
multiple sets of labels (formats)	labels defined independently of variables variables may contain one pointer to a set of labels
Scripting Language	
comments	begin line with * or //

Comments – (Stata)

Stata allows not only for notes for the dataset and each variable but also has a "Characteristics" feature which allows for the creation of named attributes for the dataset and for each variable. These metadata are stored in the .dta file which can also be exported into an XML format. Here is a snippet from that XML which shows a Universe for the dataset and a Question for the variable "z" along with some notes for variable "z".

```
<expansion>
<char name='Universe' vname='_dta'>made up numbers</char>
<char name='Question' vname='z'>What&apos;s up</char>
<char name='note1' vname='z'>This is a note for variable Z, It could have lots of stuff</char>
<char name='note0' vname='z'>2</char>
<char name='note2' vname='z'>There could also be a second note</char>
</expansion>
```

The Stata XML format for the .dta file contains both data and metadata, including value labels defined but not referenced by any variable. Transformation of the XML format would be a convenient route to a DDI file.

Issues / Restrictions – (Stata)

Since there may not be links from variables to all of the value labels, some information from beyond the normal Stata dataset may be required. In particular, Stata allows for the definition of value labels independently of variables and only allows for one link from a variable to a set of labels. Corresponding sets of labels might be defined in multiple languages, for example, and then linked from the variable as needed. DDI allows for multiple references, but some information beyond the normal Stata dataset would be needed to make these links.

This external information could be embedded in structured comments in a Stata .do file or through an interactive DDI editor working in conjunction with the importation of metadata available in a Stata dataset. It would also be possible to define a particular characteristic, e.g., named "ValueLabels", to contain a structured list of the labels applicable to the variable. This would be stored in the dataset and the only necessary external information would be the name of the characteristic. Alternatively, value labels could be treated analogously to Stata notes, where the characteristic "valuelabels0" would contain a count of the number of sets of labels and the name of the nth set of labels would be stored in characteristic valuelabels<n>.

Stata has 27 internal values for missing, denoted as ".", and ".a" to ".z". The internal representation of these values is not within the range of valid values. This is a different way of representing missing than that of SPSS, where otherwise valid values or a range and a set of valid values are tagged as missing. It is not clear how to represent these multiple values in DDI.

SAS

Metadata	Where to Find it
Dataset	
name	memname in dictionary.tables
label	memlabel in dictionary.tables
date created	crdate in in dictionary.tables
date modified	modate in dictionary.tables
encoding	encoding in dictionary.tables
VARIABLES	
name	name in dictionary.columns
basic data type	type in dictionary.columns
display format	format in dictionary.columns. Also can serve as user defined value labels
input format	informat in dictionary.columns
position	varnum in dictionary.columns
label	label in dictionary.columns
scale	????
decimal positions	format in dictionary.columns.
specific type	format in dictionary.columns.
precision	????
length	length in dictionary.columns. For numeric variables an indicator of precision
can be transcoded	transcode in dictionary.columns. Indicates whether characters can be automatically reencoded
sorted by	sortedby in dictionary.columns
VALUES	
missing	multiple, see below
multiple distinct system missing	multiple system missing values . .a .b …. .z and ._ These values may be labeled.
values as missing	dynamically through formats
ranges missing	dynamically through formats
values can be labeled	dynamically through formats
ranges can be labeled	dynamically through formats
multiple sets of labels (formats)	labels defined independently of variables. Variables may contain one pointer to a set of labels
INTEGRITY CONSTRAINTS	
range restrictions	through check constraint
list restrictions	through check constraint
foreign key	through foreign key constraint
Scripting Language	
comments	begin with * end with ; OR begin with /* end with */

Comments – (SAS)

Wackerow & Hoyle have written SAS programs to extract metadata from SAS datasets and produce DDI3.0 files. Wackerow's approach used an ODS tagset to generate the DDI (SAS_ODS_DDI_Tagset.sas). Hoyle's used a combination of DATA step and SQL code (SAS2DDI3.sas – see: http://www.ipsr.ku.edu/ksdata/sashttp/SGF2008/). A paper on both approaches is available[2].

Issues / Restrictions – (SAS)

Like Stata, SAS is capable of defining value labels (SAS user defined formats) independently from variables. An independent source of information – structured comments, SAS Macros, or a DDI editor would be necessary to describe the links between variables and all of the formats relevant to them. Unlike Stata, SAS formats can also map ranges of values into labels. It is not clear that this can currently be represented in DDI.

SAS formats are not stored in the dataset. They can be, and often are, generated on the fly from SAS code, or they can be stored in a "catalog" in a "library" from which they can be referenced by multiple programs. When SAS is running, the set of available formats can be exported into a "control" dataset referred to as a "CNTLIN" dataset for import or a "CNTLOUT" dataset for export. This dataset in a sense recreates the labeling environment for a particular dataset. Several programs which can import SAS data have the capability to use a SAS dataset and an associated CNTLIN dataset together.

Like Stata, SAS has internal values for missing numeric data, denoted as ".", and ".a" to ".z". SAS includes an additional missing value "._". The internal representation of these values is not within the range of valid values. This is a different way of representing missing than that of SPSS, where otherwise valid values or a range and a set of valid values are tagged as missing. It is not clear how to represent these multiple values in DDI.

SAS, with an SQL implementation, is also capable of working with a complex set of relations among multiple tables, including foreign key integrity constraints. We have not addressed how such a complex set of tables and relationships might be represented in DDI in this paper. One such use might be to extend the schema of the internal DICTIONARY.COLUMNS table which contains all the metadata about all columns in currently accessible datasets. Additional columns in such a dataset would correspond to the user defined attributes in SPSS or JMP or to the variable characteristics in Stata. A serious drawback would be that this would be a separate table from the dataset, with the possibility of the two being separated at some point.

[2] http://www2.sas.com/proceedings/forum2008/137-2008.pdf

JMP

Metadata	Where to Find it
Dataset	
name	dataset name
script	named script in data table
VARIABLES	
name	variable name
basic data type	Column Info - Properties - Data Type
display format	Column Info - Properties- SAS Format
position	Order of columns in table
label	Column Info - SAS Label
decimal positions	Column Info - Properties - SAS Format
specific type	Column Info - Properties - SAS Format
measurement units	Column Info - Properties - Units OR Format
weight	cols - Preselect Role
measurement level	Column Info - Modeling Type
role	Column Info - Properties - Design Role
user defined attributes	Column Info - Properties - Other
notes	Column Info - Properties - Notes
VALUES	
missing	internal missing value
values can be labeled	Column Info – Properties – Value Labels
ranges can be labeled	Column Info - Properties - Allow Ranges
value colors	Column Info - Properties - Value Colors
INTEGRITY CONSTRAINTS	
range restrictions	Column Info - Properties - Range Check
list restrictions	Column Info - Properties - List Check
Scripting Language	
comment	begin line with // OR begin comment with /* and end with */

Comments – (JMP)

All of the properties of a table can be retrieved through the JMP Scripting Language (JSL) with the "Get Script" message. The following JSL writes out a JSL script that will create the table "SimpleTable", with all metadata. This script could be parsed to transform all the data and metadata in the table to DDI. Note the "Universe" property for the column "Name" and the value labels for "Sex". Also note the table property "Bivariate". This contains a script for fitting Height by Sex. Jmp allows scripts that recreate analyses to be saved in the data table.

```
dt=Open("C:/junk/SimpleTable.jmp");
dt<<Get Script;
```

This is the resultant script:

```
New Table( "SimpleTable",
    Add Rows( 4 ),
    New Property( "Bivariate",
          Bivariate(
                Y( :Height ),
                X( :Sex ),
                Fit Mean( {Line Color( "Red" )} ),
                Density Ellipse( 0.95, {Line Color( "Green" ), Line Style( Smooth )} ),
                SendToReport(
                      Dispatch( {}, "Fit Mean ", OutlineBox, Close( 0 ) ),
                      Dispatch( {}, "Correlation ", OutlineBox, Close( 0 ) )
                )
          )
    ),
    New Column( "Name",
          Character,
          Nominal,
          Set Values( {"Joe", "Fran", "Bill", "Jill"} )
    ),
    New Column( "Height",
          Numeric,
          Continuous,
          Format( "Best", 10 ),
          Set Property( "Notes", "Represents Height in Inches" ),
          Set Property( "Universe", All People ),
          Set Values( [72, 64, 74, 60] )
    ),
    New Column( "Sex",
          Numeric,
          Continuous,
          Format( "Best", 10 ),
          Set Values( [1, 2, 1, 2] ),
          Value Labels( {1 = "Male", 2 = "Female"} ),
          Use Value Labels( 1 )
    ),
    Set Row States( [0, 0, 0, 1] )
)
```

Issues / Restrictions – (JMP)

As with the other packages there would also be the possibility of using structured comments to capture metadata.

Triple-S

Metadata	Where to Find It (Xpath)
Dataset	
Name	/sss/survey/name
Version	/sss/survey/version
Title	/sss/survey/title
VARIABLES	
Name	/sss/survey/record/variable/name
basic data type	/sss/survey/record/variable/@format = literal OR numeric
Label	/sss/survey/record/variable/label
specific type	/sss/survey/record/variable/@type = single OR multiple OR quantity OR character OR logical OR data OR time
Weight	/sss/survey/record/variable/@use = weight
filter (points to logical variable. True if this variable is available for this case	/sss/survey/record/variable/filter
VALUES	
values can be labeled	coded value = /sss/survey/record/variable/values/value/@code label = /sss/survey/record/variable/values/value
INTEGRITY CONSTRAINTS	
range restrictions	/sss/survey/record/variable/values/range
Comments	
Structured comments possible	<!--*comment_text*--> can be used anywhere (after the initial <?xml …> declaration

Comments – (Triple-S)

Triple-S is an XML format for moving surveys between survey packages on various hardware and software platforms. Triple-S appears to be primarily aimed at interchange of data, without the level of descriptive metadata possible in DDI.

Issues / Restrictions – (Triple-S)

Triple-S has the capability to represent a complex hierarchy with links to multiple data files. As with relational databases these extended features will not be addressed here.

StatTransfer

Metadata	Where to Find it
Dataset	
name	
VARIABLES	
name	
basic data type	
display format	
position	
label	
decimal positions	
specific type	
user defined attributes	can append some user defined attributes to variable label
notes	
VALUES	
missing	
multiple distinct system missing	e.g., from SAS or Stata
values as missing	e.g., from SPSS
ranges missing	reads SPSS missing specification, can map to multiple system missing
values can be labeled	
ranges can be labeled	imports SAS formats
Scripting Language	
structured comments possible	comments preceded by // in command language

Comments – (StatTransfer)

A program like StatTransfer could be a good tool for capturing metadata from datasets like those considered in this paper. It would even be possible for it to write a DDI file directly. StatTransfer, for example, currently writes Triple-S XML. In the absence of direct DDI capability, something like StatTransfer can be used to convert to a format like SPSS or SAS for which conversion tools currently exist. StatTransfer has no external storage format of its own; it is for conversion purposes only.

Issues / Restrictions – (StatTransfer)

Conversion from one package to another carries a few risks. As an example we ran a quick test of using StatTransfer to convert from a Stata file to several other formats. The Stata file had multiple sets of labels for the variable sex (English and French, with the English form attached to the variable) and a characteristic "Question". When converting to SAS, only the labels which were assigned to a variable were transferred. When converting to SPSS the "Question" characteristic was not set as a custom attribute. The StatTransfer documentation states that custom properties can be appended to variable labels.

These R objects are supported by StatTransfer: 2 dimensional matrices, vectors, factors, and dataframes.

Relational Databases

As mentioned in the discussion of SAS, relational databases are capable of working with a complex set of relations among multiple tables, including foreign key integrity constraints. We have not addressed how such a complex set of tables and relationships might be represented in DDI in this paper. It would be noted, however, that DDI metadata can be represented in a relational model.

Metadata from single tables can be exported. Furthermore the relationship between two tables (for example, person to household relationship) can be exported and represented in DDI.

Example: Microsoft Access

Metadata	Where to Find it
Dataset	
name	table name
label	Table Properties…Description
script	Database Tools.. Visual Basic
VARIABLES	
name	Field Name
basic data type	Data Type
display format	Format
position	OrdinalPosition
label	Description and Caption
decimal positions	Decimal Places
specific type	Field Size
measurement units	Format
VALUES	
missing	one internal value
values can be labeled	by relation with another table
INTEGRITY CONSTRAINTS	
range restrictions	Validation Rule
list restrictions	Validation Rule
restriction by expression	Validation Rule
foreign key	relationships - enforce referential integrity
Scripting Language	
structured comments possible	in VBA modules
Scripting Language	Visual Basic
comments	begin line with single quote

Comments – (MS Access)

As a relational database, Access is capable of representing complex sets of relationships among multiple tables. Multiple sets of value labels, instead of being represented as labels (as in Stata) or formats (as in SAS) can be stored in an Access database as views (queries) on related tables. Writing software to extract these metadata from the arbitrary Access database would be a difficult task.

Issues / Restrictions – (MS Access)

As with the other packages there would be the possibility of using structured comments in the scripting language for Access (Visual Basic for Applications – [VBA]) to capture metadata.

SOFTWARE COMPARISON

Metadata Element	CSV File	Excel	R	SPSS	Stata	SAS	JMP	MS Access	Triple-S	Stat Transfer	DDI Parent Element	DDI Element	Note
Dataset											ddi:DDIInstance	s:StudyUnit	possible usage
name	-	x	x	x	x	x	x	x	x	x	l:LogicalProduct	l:LogicalProductName	
label	-	-	-	x	x	x	x	x	x		l:LogicalProduct	r:Label	
user defined attributes	-	-	-	x	-	-	-	-	-		l:LogicalProduct	r:Description	DDI: or a more specific element
Script stored with data	-	x	-	-	-	-	x	x	-	-	l:LogicalProduct	r:Description	
VARIABLES											l:VariableScheme	l:Variable	
name	x	x	x	x	x	x	x	x	x	x	l:Variable	l:VariableName	
basic data type	x	x	x	x	x	x	x	x	x	x	l:Representation	l:DateTimeRepresentation OR l:NumericRepresentation OR l:TextRepresentation additional possibility: r:RecommendedDataType	numeric, character, DateTime DDI: this data type definition is application independent
											m4:DataItem	m4:ProprietaryDataType	DDI: the proprietary data type is application dependent.
display format	-	x	x	x	x	x	x	x	-	x	l:Representation	r:GenericOutputFormat	DDI: the generic output format is application independent
											m4:DataItem	m4:ProprietaryOutputFormat	DDI: the proprietary output format is application dependent.

SOFTWARE COMPARISON – CONTINUED (VARIABLES)

Metadata Element	CSV File	Excel	R	SPSS	Stata	SAS	JMP	MS Access	Triple-S	Stat Transfer	DDI Parent Element	DDI Element	Note
position	x	x	x	x	x	x	x	x	x	x	p:PhysicalDataProduct OR ds:RecordSet	order of l:Variable in l:VariableScheme (logical level) and p:Dataitem in p:RecordLayout OR order of ds:VariableReference in ds:VariableOrder (physical level)	The sequence of the variables can be different on the logical (definition) level and the physical representation level.
label	-	-	-	x	x	x	x	x	x	x	l:Variable	r:Label	
scale	-		-			?		-	-	-	l:NumericRepresentation	@scale	use in SAS unclear
decimal positions	x	x	x	x	x	x	x	x	-	x	l:NumericRepresentation	@decimalPositions	in format specifications
specific type	x	x	x	x	x	x	x	x	x	x	l:Representation	l:DateTimeRepresentation @type OR l:NumericRepresentation @type	DateTime, Date, Time, Integer, Float, etc.
measurement units	-	-	-	-	-	-	x	x	-	-	l:Variable	r:AnalysisUnit	display formats such as Euros
weight	-	-	-	x	-	-	x	-	x	-	l:Variable	@isWeight	
measurement level	-	-	x	x	-	-	x	-	-	-	l:CodeRepresentation	@classificationlevel	e.g., Nominal, ordinal
role	-	-	-	-	-	-	x	-		-	l:Representation	l:Role	
user defined attributes	-	-	x	x		-	x	-	-	x	l:Variable	r:Description	DDI: or a more specific element
notes	-	-	-	-	x	-	x	-	-		l:LogicalProduct	r:Note	A DDI note has a reference to the variable to which the note is referring (r:Relationship)
filter	-	-	-	-	-	-	-	-	x	-			points to another (logical) variable which indicates whether this variable is available for this case

SOFTWARE COMPARISON – CONTINUED

Metadata Element	CSV File	Excel	R	SPSS	Stata	SAS	JMP	MS Access	Triple-S	Stat Transfer	DDI Parent Element	DDI Element	Note
VALUES													
missing	x	x	x	x	x	x	x	x	-	x	l: CodeRepresentation OR l:DateTimeRepresentation OR l:ExternalCategoryRepresentation OR l:NumericRepresentation OR l:TextRepresentation	@missingValue	single missing value
multiple distinct system missing	-	-	-	-	x	x		-	-	x	l:Variable	r:Description	DDI: a dedicated element for system missing is not yet available in DDI
multiple values as missing	-	-	-	x	-	*	-	-	-	x	l:Category	@missing	DDI: a category referenced by a code can have an indicator if missing
ranges missing	-	-	-	x	-	*	-	-	-	x	l:Category	@missing	DDI: ranges must be translated in single missing values
values can be labeled	-	-	-	x	x	x	x	x	x	x	l:CodeRepresentation	l:CategorySchemeReference	Access: foreign key
ranges can be labeled	-	-	-	-	-	x	x	-	-	x	l:NumericRepresentation	l:NumberRange	no associated label in DDI
multiple sets of labels (formats)	-	-	-	-	x	x	-	-	-	-	-	-	DDI: This would be represented by multiple variables, which refer to the same physical location
value colors	-	-	-	-	-	-	x	-	-	-	l:Variable	r:Description	

SOFTWARE COMPARISON – CONTINUED

Metadata Element	CSV File	Excel	R	SPSS	Stata	SAS	JMP	MS Access	Triple-S	Stat Transfer	DDI Parent Element	DDI Element	Note
INTEGRITY CONSTRAINTS													
range restrictions	-	x	-	-	-	x	x	x	x	-	l:NumericRepresentation	r:NumberRange	
list restrictions	-	x	-	-	-	x	x	x	-	-	l:CodeScheme	l:Code	
restriction by expression	-	-	-	-	-	x	-	x	-	-	l:Variable	r:Description	DDI: a dedicated element for system missing is not yet available in DDI
foreign key	-	-	-	-	-	x	-	x	-	-	l:LogicalProduct	l:DataRelationship	
Scripting Language													
structured comments possible	-	x	x	x	x	x	x	x	x	x			DDI: Keywords of structured comments can be mapped to the content of simple DDI elements.
Software-specific													
Name, version, description		x	x	x	x	x	x	x		x	m4:ProprietaryRecordLayout	m4:Software	
Information on the data file		x	x	x	x	x	x	x		x	m4:ProprietaryRecordLayout/ m4:ProprietaryInfo	m4:ProprietaryProperty	DDI: user-defined property
Information on variables		x	x	x	x	x	x	x		x	m4:DataItem/m4:ProprietaryInfo	m4:ProprietaryProperty	DDI: user-defined property

General possibilities and difficulties

One type of metadata embedded in the datasets of most of these programs is associated with data formats. If a variable is formatted as "Euros," it carries the information both that the values represent currency and that they are measured in Euros. Good techniques for metadata capture will enumerate the possible formats for each software package and include those implicit metadata.

We have discussed using various techniques to represent metadata not available in the native format of the package's dataset, structured comments, Stata characteristics, SAS macros, relational tables, and SPSS user defined attributes. In all of these cases it is likely that for some metadata elements these will be unsatisfactory. Elements containing long XHTML structured content will be awkward to enter as text and in many cases there are restrictions on string lengths which would exclude desired content. An editor that allows a visual representation of the markup (headers, tables, etc.) will be important.

Structured comments and/or notes, parsing code

Some form of structured comments is possible in every one of the packages or data formats we analyzed. Comments could be used to embed additional metadata as follows. The user writes specially-formatted comments in the code of the related statistical package itself. A special application parses these comments and transforms them into documentation in HTML or PDF format. An example tool for this purpose is ROBODoc[3]. The idea comes from documenting program code inline like the widely used Javadoc application. This approach can be adapted in the way that the application transforms this information into DDI fragments. These fragments can be added to existing DDI documents. This approach would often fit into existing work flows when users already add comments to the code in the statistical package. The change is that the comments must contain known keywords and must be wrapped by characters with special meanings so the parser can recognize them.

Example for SPSS:

```
***v* Variables/DerivedVariables
 * DESCRIPTION
 * This section is on derived variables;
***.
***v* DerivedVariables/w101_new
 * NAME
 * w101_new
 * DESCRIPTION
 * w101_new is a derived variable from w101;
 * It has the original value from w101
 * when w102 is equal 1
 * otherwise it has the value 5;
 * USED VARIABLES
 * w101, w102
 * SOURCE
**.
compute w101_new = 5 .
if ( w102 = 1 ) w101_new = w101 .
```

[3] http://www.xs4all.nl/~rfsber/Robo/robodoc.html

Crosswalk Software

StatTransfer or comparable software can serve as a crosswalk between these statistical packages or the triple-S XML format (http://www.triple-s.org). In some cases this may result in a loss of metadata when converting to a format incapable of representing metadata present in the original format. Examples include moving a dataset from one allowing multiple system missing values (e.g., Stata or SAS) to a system allowing only one missing value (e.g., R), or moving data from a system allowing multiple variables to reference a single set of value labels (e.g., Stata or SAS) to one in which value labels are tied directly to the variable (e.g., SPSS or JMP).

DDI Issues

We have identified a few instances where representing content in one or more of these packages in DDI is problematic. Examples include multiple system missing, labels for ranges of values, and integrity constraints such as arbitrary where clauses and foreign key constraints.

OUTLOOK / CONCLUSION

One advantage of a structured comments approach is that it could be somewhat software independent. While there is not a common format for comments across all packages, translation should be fairly easy.

DDI has the richest metadata set with some restrictions. Therefore it could act as exchange format between all of these programs. A desirable development would be for these programs to add DDI export and import of the metadata they do hold, but the most likely scenario in the near term would be for a program like StatTransfer to add DDI export and import.

An even better solution would be for these packages to add tools that bring more focus to the complete data life cycle, with tools for metadata collection beginning at the design stage, not just at analysis stage. As a collection none of these packages have all of the tools to support the complete data life cycle. Most have no structured place for nor the tools to manage study-level metadata – e.g., purpose, funding, author, sampling, concepts and more. This is a focus of upcoming tools such as the DDI Editor-Lite (see: http://www.ddialliance.org/node/408 or perhaps Algenta Colectica (see: http://www.colectica.com/).

One of the packages we looked at, SAS JMP, has interactive tools for design of experiments (DOE) as part of its basic set of features. There are also R packages with similar functionality (see: http://cran.r-project.org/web/views/ExperimentalDesign.html). One package, SPSS, has the capability to add user defined attributes of variables to its metadata entry forms for variables. User defined attributes are also present in R and Stata, but are accessible only through their programming languages. One package, SAS Enterprise Guide (EG), is capable of capturing, modifying, and rerunning annotated flow diagrams of the analysis process, although its only export facility appears to be exporting SAS code. SPSS, Stata, and R have a more limited capability of managing and archiving the analysis process. They can capture journal files of some of the steps taken interactively in a session.

External programs to extract metadata from any of these tools will be useful, but are not the same thing as a single integrated environment in which to work for the whole data lifecycle. In the end though, it is likely that those doing research will continue to use the tools they feel will make them the most productive – which in

many cases is the set of tools with which they are most familiar. External tools that integrate nicely with those familiar tools will be most useful.

REFERENCES – MORE INFORMATION

CSV files -	http://en.wikipedia.org/wiki/Comma-separated_values
Microsoft Excel -	http://office.microsoft.com/excel
R Language -	http://www.r-project.org/
IBM SPSS Statistics-	http://www.spss.com/statistics/
Stata -	http://www.stata.com/
The SAS System -	http://www.sas.com/
SAS JMP -	http://www.jmp.com/
Microsoft Access -	http://office.microsoft.com/access
Triple S -	http://www.triple-s.org/
Stat Transfer -	http://www.stattransfer.com/
DDI -	http://www.ddialliance.org/

APPENDIX A

The paper is one of several papers which are the outcome of a workshop held at Schloss Dagstuhl - Leibniz Center for Informatics in Wadern, Germany, November 2-6, 2009.

Workshop title:
Workshop on Implementation of DDI3 - Advanced Topics

Organizers:
Arofan Gregory (Open Data Foundation, Tucson, Arizona, USA)
Wendy Thomas (Minnesota Population Center, University of Minnesota, USA)
Mary Vardigan (Inter-university Consortium for Political and Social Research [ICPSR], University of Michigan, USA)
Joachim Wackerow (GESIS, Leibniz Institute for the Social Sciences, Germany)
Link: http://www.dagstuhl.de/09452

This series was edited by Michelle Edwards, Larry Hoyle and Mary Vardigan.

The authors of the paper would like to acknowledge others who participated in this workshop.

Alerk Amin, CentERdata, Tilburg University, the Netherlands
Michelle Edwards, University of Guelph, Canada
Bryan Fitzpatrick, Rapanea Consulting, United Kingdom
Oliver Hopt, GESIS, Leibniz Institute for the Social Sciences, Bonn, Germany
Larry Hoyle, Institute for Policy and Social Research, University of Kansas, USA
Sanda Ionescu, Inter-university Consortium for Political and Social Research (ICPSR), University of Michigan, USA
Jannik Jensen, Dansk Data Archive (DDA), Denmark
Uwe Jensen, GESIS, Leibniz Institute for the Social Sciences, Köln, Germany
Mari Kleemola, Finnish Social Science Data Archive (FSD), University of Tampere, Finland
Dan Kristiansen, Dansk Data Archive (DDA), Denmark
Agostina Martinez, University of Cambridge, United Kingdom
Martin Mechtel, Institute for Educational Progress, Humboldt-Universität zu Berlin, Germany
Olof Olsson, Swedish National Data Service (SND), Sweden
Ørnulf Risnes, Norwegian Social Science Data Services (NSD), Norway
Wolfgang Zenk-Möltgen, GESIS, Leibniz Institute for the Social Sciences, Köln, Germany

APPENDIX B