



HOW TO EFFECTIVELY DEAL WITH HARDCODING AND CDISC CONTROLLED TERMINOLOGY IN CLINICAL STUDIES



Paper DH11

Lennert van der Zee, OCS Consulting, Netherlands

AGENDA



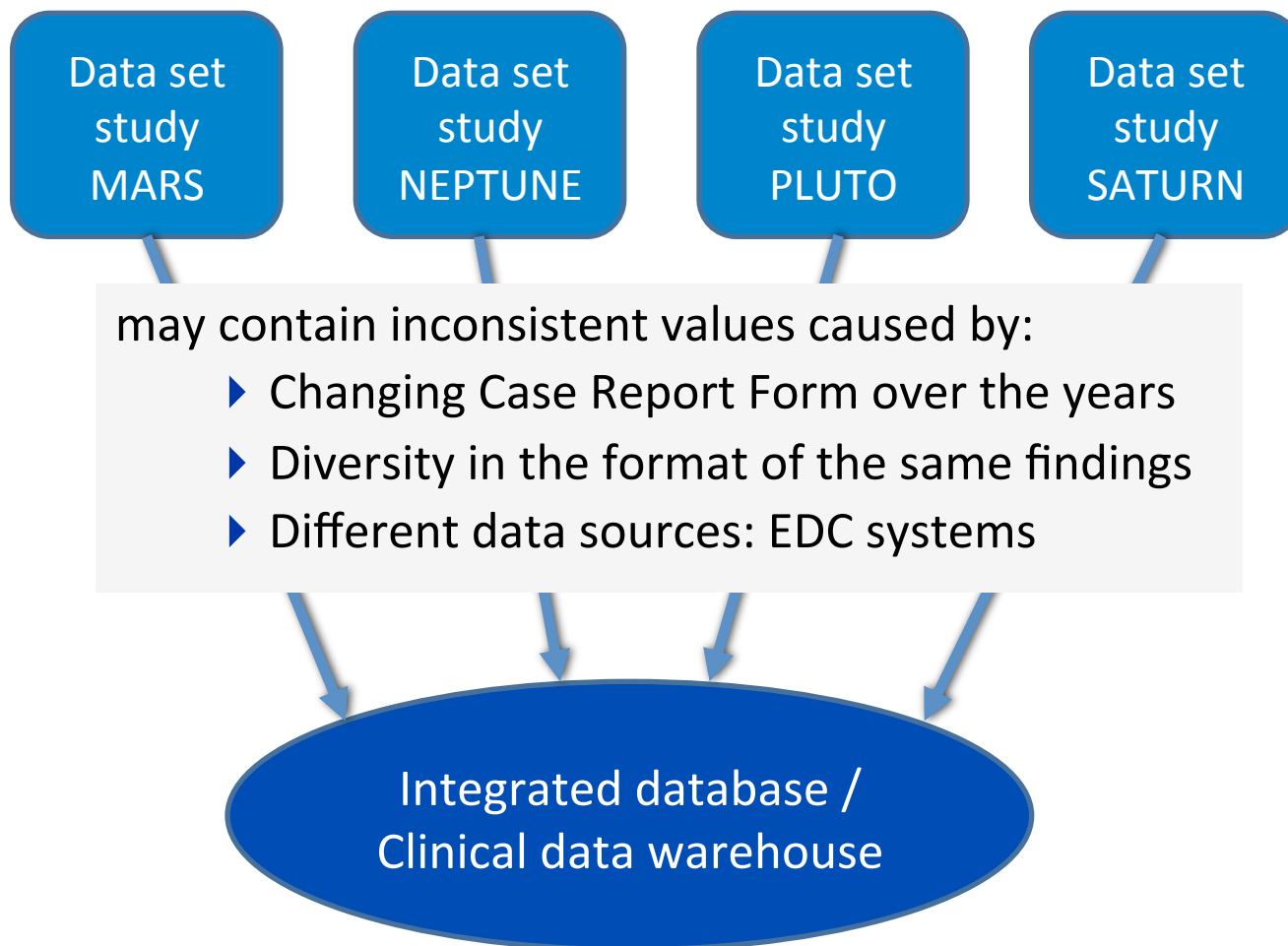
- ▶ Background
- ▶ Undesirable way of hardcoding
- ▶ Updating values of variables by using Controlled Terminology
 - ▶ 2 Manual steps
 - ▶ 6 Macro steps
- ▶ Extra feature
- ▶ Conclusion
- ▶ Questions

SAS macros can be downloaded from:

<http://ocs-consulting.nl/phuse2016-dh11>

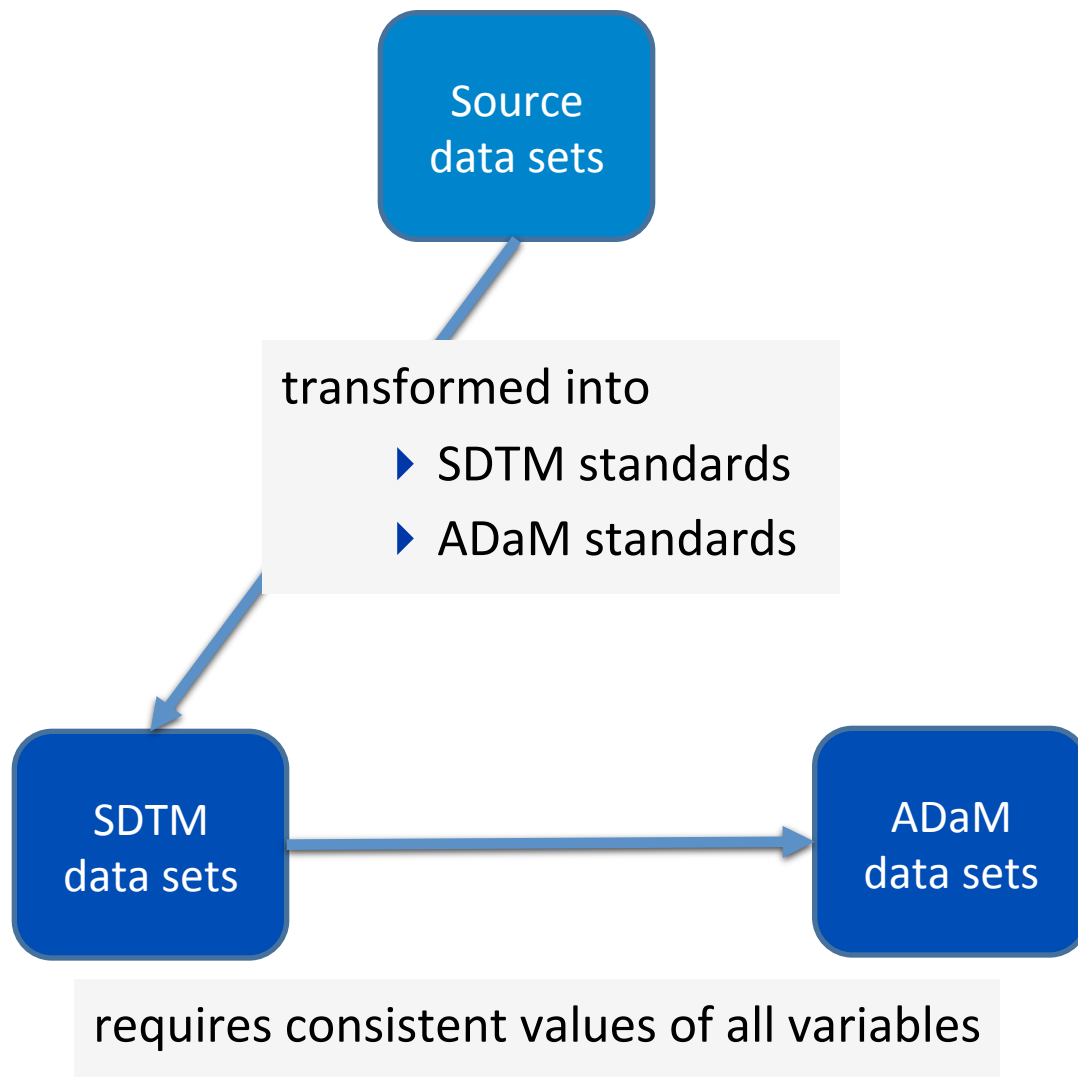


BACKGROUND



requires consistent values of all variables across studies

BACKGROUND



BACKGROUND



How to solve inconsistent values of variables?



- ▶ Applying hardcoding in the individual SAS programs



- ▶ In one document:
Collecting all original values of variables and their associated values that are converted to uniform values
- ▶ In every individual SAS program:
Including one or more SAS macros calls in which values are read from the document and the values of the variables are updated in the data sets

Developed SAS macro **UPDATE_VALUES**

BACKGROUND



Description of the effective created SAS macro **UPDATE_VALUES**:

- ▶ Reading a single Excel spreadsheet with value updates of variables
- ▶ Updating values of variables
- ▶ Ensuring that all values are consistent and comply with CDISC Controlled Terminology
- ▶ Illustrated by using sample data



BACKGROUND



An example of an integrated database for Adverse Events

STUDYID	SUBJID	AESEQ	AETERM	AESEV	AEREL	All other variables
MARS	1001	1	NAUSEA	MILD SEVERITY	UNLIKELY REL	
MARS	1002	1	DIZZINESS	MODERATE SEVERITY	POSSIBLE REL	
MARS	1003	1	FEVER	MODERATE SEVERITY	UNLIKELY REL	
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD	UNRELATED	
NEPTUNE	1002	1	COUGH	SEV	PROBABLE	
NEPTUNE	1003	1	VOMITING	MOD	DEFINITE	
PLUTO	1001	1	COLD	1	3	
PLUTO	1001	2	NAUSEA	1	1	
PLUTO	1002	1	DIARRHOEA	1	2	
SATURN	1001	1	LOSS OF APPETITE	SEVERE	PROBABLY	
SATURN	1001	2	HEADACHE	MILD	UNLIKELY	
SATURN	1001	3	NAUSEA	MILD	NOT RELATED	

- ▶ Updating values of variables AESEV and AEREL
a demonstration of value updates of:
 - ▶ Variable AESEV
 - ▶ Study MARS

HARDCODING



STUDYID	SUBJID	AESEQ	AETERM	AESV	AEREL	
MARS	1001	1	NAUSEA	MILD SEVERITY	UNLIKELY REL	MILD
MARS	1002	1	DIZZINESS	MODERATE SEVERITY	POSSIBLE REL	MODERATE
MARS	1003	1	FEVER	MODERATE SEVERITY	UNLIKELY REL	MODERATE
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD	UNRELATED	
NEPTUNE	1002	1	COUGH	SEV	PROBABLE	
NEPTUNE	1003	1	VOMITING	MOD	DEFINITE	
PLUTO	1001	1	COLD	1	3	
PLUTO	1001	2	NAUSEA	1	1	
PLUTO	1002	1	DIARRHOEA	1	2	
SATURN	1001	1	LOSS OF APPETITE	SEVERE	PROBABLY	
SATURN	1001	2	HEADACHE	MILD	UNLIKELY	
SATURN	1001	3	NAUSEA	MILD	NOT RELATED	

All other variables

- ▶ A still regularly used method

```

if upcase(studyid)="MARS" then
do;
    if      upcase(aesev)="MILD SEVERITY"      then aesev="MILD";
    else if upcase(aesev)="MODERATE SEVERITY" then
aesev="MODERATE";
end;
    
```

HARDCODING



Hardcoding is undesirable:

- ▶ Lack of flexibility to revise values of variables
- ▶ Very time-consuming:
 - ▶ Revising values of variables
 - ▶ Validating programs

CONTROLLED TERMINOLOGY



Most effective method for updating values of variables:

- ▶ Created SAS macro **UPDATE_VALUES**

Functioning of SAS macro **UPDATE_VALUES**:

- ▶ Steps done by the user:
2 manual steps
- ▶ Steps done by the macro:
6 macro steps

CT – MANUAL STEP 1



All unique values of variables per study are collected and entered in an Excel spreadsheet

Required items in Excel spreadsheet for all value updates:

- ▶ Study name
- ▶ Name or title that covers the contents of the variable
- ▶ Original value of the variable
- ▶ New value updates of the variable

STUDY	TERM	ORIGINAL VALUE	NEW VALUE

Choose generic name so that values can be used for multiple variables from different data sets

stored in a generic folder within the project

CT – MANUAL STEP 1



STUDYID	SUBJID	AESQ	AETERM	AESV	ORG	AEREL
MARS	1001	1	NAUSEA	MILD SEVERITY		UNLIKELY REL
MARS	1002	1	DIZZINESS	MODERATE SEVERITY		POSSIBLE REL
MARS	1003	1	FEVER	MODERATE SEVERITY		UNLIKELY REL
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD		UNRELATED
NEPTUNE	1002	1	COUGH	SEV		PROBABLE
NEPTUNE	1003	1	VOMITING	MOD		DEFINITE
PLUTO	1001	1	COLD	1		3
PLUTO	1001	2	NAUSEA	1		1
PLUTO	1002	1	DIARRHOEA	1		2

All other variables

MILD

MODERATE

MODERATE

Adverse Events
data set

SEVERITY

recommended to use
the complete CDISC code list,
if new values should meet
CDISC Controlled
Terminology

STUDY	TERM	ORIGINAL VALUE	NEW VALUE
MARS	RELATIONSHIP	POSSIBLE REL	POSSIBLY RELATED
MARS	RELATIONSHIP	UNLIKELY REL	UNLIKELY RELATED
NEPTUNE	RELATIONSHIP	DEFINITE	DEFINITELY RELATED
NEPTUNE	RELATIONSHIP	UNRELATED	NOT RELATED
NEPTUNE	RELATIONSHIP	PROBABLE	PROBABLY RELATED
PLUTO	RELATIONSHIP	1	NOT RELATED
PLUTO	RELATIONSHIP	3	POSSIBLY RELATED
PLUTO	RELATIONSHIP	2	UNLIKELY RELATED
MARS	SEVERITY	MILD SEVERITY	MILD
MARS	SEVERITY	MODERATE SEVERITY	MODERATE
NEPTUNE	SEVERITY	MOD	MODERATE
NEPTUNE	SEVERITY	SEV	SEVERE
PLUTO	SEVERITY	1	MILD

SEVERITY

MILD

MODERATE

Excel spreadsheet
update_values.xlsx

CT – MANUAL STEP 2



Parameters are entered in SAS macro **UPDATE_VALUES**:

- ▶ Input SAS data set name
- ▶ Output SAS data set name
- ▶ Study name or protocol number
- ▶ Name or title that covers the contents of the variable
- ▶ Variable name containing original values
- ▶ Variable name containing new value updates

```
%update_values (inset =  
                ,outset=  
                ,study =  
                ,term  =  
                ,orgvar=  
                ,newvar=  
                ) ;
```

CT – MANUAL STEP 2



STUDYID	SUBJID	AESQ	AETERM	AESEV_ORG	AEREL	
MARS	1001	1	NAUSEA	MILD SEVERITY	UNLIKELY REL	MILD
MARS	1002	1	DIZZINESS	MODERATE SEVERITY	POSSIBLE REL	MODERATE
MARS	1003	1	FEVER	MODERATE SEVERITY	UNLIKELY REL	MODERATE
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD	UNRELATED	
NEPTUNE	1002	1	COUGH	SEV	PROBABLE	
NEPTUNE	1003	1	VOMITING	MOD	DEFINITE	
PLUTO	1001	1	COLD	1	3	
PLUTO	1001	2	NAUSEA	1	1	
PLUTO	1002	1	DIARRHOEA	1	2	

All other variables

SEVERITY

Rename to variable
AESEV_ORG

```
%update_values(inset =work.ae01
               ,outset=work.ae02
               ,study =MARS
               ,term  =SEVERITY
               ,orgvar=AESEV_ORG
               ,newvar=AESEV_NEW
               );
```

AESEV_ORG

AESEV_NEW

STUDY	TERM	ORIGINAL VALUE	NEW VALUE
MARS	RELATIONSHIP	POSSIBLE REL	POSSIBLY RELATED
MARS	RELATIONSHIP	UNLIKELY REL	UNLIKELY RELATED
NEPTUNE	RELATIONSHIP	DEFINITE	DEFINITELY RELATED
NEPTUNE	RELATIONSHIP	UNRELATED	NOT RELATED
NEPTUNE	RELATIONSHIP	PROBABLE	PROBABLY RELATED
PLUTO	RELATIONSHIP	1	NOT RELATED
PLUTO	RELATIONSHIP	3	POSSIBLY RELATED
PLUTO	RELATIONSHIP	2	UNLIKELY RELATED
MARS	SEVERITY	MILD SEVERITY	MILD
MARS	SEVERITY	MODERATE SEVERITY	MODERATE
NEPTUNE	SEVERITY	MOD	MODERATE
NEPTUNE	SEVERITY	SEV	SEVERE
PLUTO	SEVERITY	1	MILD

CT – MACRO STEP 1



All data stored in the Excel spreadsheet are read:

- ▶ General and separate SAS macro that only reads data from Excel spreadsheets

Required parameters for SAS macro **READ_XLSX_FILE**:

- ▶ Pathname where the Excel spreadsheet is stored
- ▶ Filename of the Excel spreadsheet
- ▶ Sheet name in the Excel spreadsheet
- ▶ Output SAS data set name

```
%read_xlsx_file(pathname=C:\PhUSE2016\DH11\Metadata  
                ,filename=update_values.xlsx  
                ,sheet    =Sheet1  
                ,outset   =work.updatevalues  
                );
```

CT – MACRO STEP 2



Row order in the data set should be maintained:

- ▶ Temporary variable **ROWNUM** is defined
- ▶ Variable **ROWNUM** is based on the sequence number of the records in the Adverse Events data set

```
data work.inset_sorted;
  set &inset;
  rownum+1;
run;
```

work.ae01

STUDYID	SUBJID	AESEQ	AETERM	AESEV_ORG	ROWNUM
MARS	1001	1	NAUSEA	MILD SEVERITY	1
MARS	1002	1	DIZZINESS	MODERATE SEVERITY	2
MARS	1003	1	FEVER	MODERATE SEVERITY	3
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD	4
NEPTUNE	1002	1	COUGH	SEV	5
NEPTUNE	1003	1	VOMITING	MOD	6
PLUTO	1001	1	COLD	1	7
PLUTO	1001	2	NAUSEA	1	8
PLUTO	1002	1	DIARRHOEA	1	9
SATURN	1001	1	LOSS OF APPETITE	SEVERE	10
SATURN	1001	2	HEADACHE	MILD	11
SATURN	1001	3	NAUSEA	MILD	12

AEREL + All other variables

data set
work.inset_sorted



CT – MACRO STEP 3

Type of the variables in Excel spreadsheet are determined:

- ▶ General and separate SAS macro that returns the values:
 - ▶ value "C"
if variable contains character values
 - ▶ value "N"
If variable contains numeric values

Required parameters for SAS macro **GETVARTYPE**:

- ▶ SAS data set name
- ▶ Variable that needs to be updated

```
%getvartype(dset=inset_sorted  
            ,var =&orgvar  
            );
```

AESEV_ORG

CT – MACRO STEP 3



Either value "C" or "N" is assigned to macro variable **VARTYPE**:

```
%macro getvartype(dset=, var=);  
  %let rc=%sysfunc(open(&dset, i));  
  
  %if &rc>0 %then  
    %do; %let varnum=%sysfunc(varnum(&rc, &var));  
        %let vartype=%sysfunc(vartype(&rc, &varnum));  
        %let rc     =%sysfunc(close(&rc));  
    %end;  
  %else %put WARNING: Macro getvartype cannot open data set &dset.;  
  
  &vartype  
%mend getvartype;
```

No semicolon
Value "C" or "N" is returned

```
%let vartype=%getvartype(dset=inset_sorted  
                        ,var =&orgvar  
                        );  
"C"
```



CT – MACRO STEP 4

Length of the variable **AESEV_ORG** is determined:

- ▶ Preventing truncation of the values
- ▶ Only needed if the variable is character

Required parameters for SAS macro **GETVARLENGTH**:

- ▶ SAS data set name
- ▶ Variable that needs to be updated

```
%getvarlength(dset=inset_sorted  
              ,var =&orgvar  
              );
```

AESEV_ORG



CT – MACRO STEP 4



Length value is assigned to macro variable **VARLEN**:

```
%macro getvarlength(dset=, var=);  
  %let rc=%sysfunc(open(&dset, i));  
  
  %if &rc>0 %then  
    %do; %let varnum=%sysfunc(varnum(&rc, &var));  
        %let varlen=%sysfunc(varlen(&rc, &varnum));  
        %let rc      =%sysfunc(close(&rc));  
    %end;  
  %else %put WARNING: Macro getvarlen cannot open dataset &dset.;  
  
  &varlen  
%mend getvarlength;
```

No semicolon
Length of the variable is returned

```
%if %upcase(&vartype)=C %then  
%do; %let varlen=%getvarlength(dset=inset_sorted  
                                ,var =&orgvar  
                                );  
%end;
```

50

CT – MACRO STEP 5



Required data from Excel spreadsheet with value updates is selected:

- ▶ Original values **ORIGINAL_VALUE** are assigned to variable **AESEV_ORG**
- ▶ New values **NEW_VALUE** are assigned to variable **AESEV_NEW**
- ▶ Applying macros **GETVARTYPE** and **GETVARLENGTH**

	STUDY	TERM	ORIGINAL VALUE	NEW VALUE
	MARS	RELATIONSHIP	POSSIBLE REL	POSSIBLY RELATED
	MARS	RELATIONSHIP	UNLIKELY REL	UNLIKELY RELATED
	NEPTUNE	RELATIONSHIP	DEFINITE	DEFINITELY RELATED
	NEPTUNE	RELATIONSHIP	UNRELATED	NOT RELATED
	NEPTUNE	RELATIONSHIP	PROBABLE	PROBABLY RELATED
	PLUTO	RELATIONSHIP	1	NOT RELATED
	PLUTO	RELATIONSHIP	3	POSSIBLY RELATED
	PLUTO	RELATIONSHIP	2	UNLIKELY RELATED
	MARS	SEVERITY	MILD SEVERITY	MILD
	MARS	SEVERITY	MODERATE SEVERITY	MODERATE
	NEPTUNE	SEVERITY	MOD	MODERATE
	NEPTUNE	SEVERITY	SEV	SEVERE
	PLUTO	SEVERITY	1	MILD

CT – MACRO STEP 5



```
data work.updatevalues_selected;
  set work.updatevalues(where=(      upcase(study)=upcase("&study")
                                     and upcase(term) =upcase("&term")));

  %if %upcase(&vartype)=N %then
  %do; length &orgvar 8;
        &orgvar=input(original_value, 8.0);
  %end;

  %else %if %upcase(&vartype)=C %then
  %do; %let varlen=%getvarlength(dset=&inset
                                ,var =&orgvar
                                );
        length &orgvar $ &varlen;
        &orgvar=original_value;
  %end;

  rename new_value=&newvar;
  drop term original_value;
run;
```

MARS (points to "&study")

SEVERITY (points to "&term")

work.ae01 (points to "dset=&inset")

AESEV_ORG (points to "&orgvar")

AESEV_NEW (points to "&newvar")

(ORIGINAL_VALUE) (points to "original_value")

(NEW_VALUE) (points to "&newvar")

STUDYID	AESEV_ORG	AESEV_NEW
MARS	MILD SEVERITY	MILD
MARS	MODERATE SEVERITY	MODERATE

CT – MACRO STEP 6



Data sets are merged:

- ▶ Data set with updated values are merged with Adverse Events data set based on study name and variable **AESEV_ORG**
- ▶ New data set is sorted based on temporary variable **ROWNUM**

```
proc sql;
    create table &outset(drop=rownum) as
    select      A.*
               ,B.&newvar
    from        work.inset_sorted as A
    left join   work.updatevalues_selected as B
    on          upcase(A.studyid)=upcase(B.study)
               and upcase(A.&orgvar)=upcase(B.&orgvar)
    order by   rownum;
quit;
```

Annotations in the code:

- work.ae02** points to the `work.inset_sorted` table.
- AESEV_NEW** points to the `&newvar` variable.
- AESEV_ORG** points to the `&orgvar` variable.

CT – MACRO STEP 6



Resulting data set:

- Variables **AESEV_ORG** and **AESEV_NEW** provide insight into how Controlled Terminology is applied

STUDYID	SUBJID	AESEQ	AETERM	AESEV_ORG	AESEV_NEW
MARS	1001	1	NAUSEA	MILD SEVERITY	MILD
MARS	1002	1	DIZZINESS	MODERATE SEVERITY	MODERATE
MARS	1003	1	FEVER	MODERATE SEVERITY	MODERATE
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD	
NEPTUNE	1002	1	COUGH	SEV	
NEPTUNE	1003	1	VOMITING	MOD	
PLUTO	1001	1	COLD	1	
PLUTO	1001	2	NAUSEA	1	
PLUTO	1002	1	DIARRHOEA	1	
SATURN	1001	1	LOSS OF APPETITE	SEVERE	
SATURN	1001	2	HEADACHE	MILD	
SATURN	1001	3	NAUSEA	MILD	

AESEV + All other variables

Drop variable
AESEV_ORG

Rename to variable
AESEV

EXTRA FEATURE



Values of one variable need to be updated for all studies

```
data work.updatevalues_selected;
  set work.updatevalues (where=(%if &study^=%str() %then
                                %do; upcase(study)=upcase("&study") and %end;
                                upcase(term) =upcase("&term")));

  %if %upcase(&vartype)=N %then
  %do; length &orgvar 8;
        &orgvar=input(original_value, 8.0);
  %end;

  %else %if %upcase(&vartype)=C %then
  %do; %let varlen=%getvarlength(dset=&inset
                                ,var =&orgvar
                                );
        length &orgvar $ &varlen;
        &orgvar=original_value;
  %end;

  rename new_value=&newvar;
  drop term original_value;

run;
```

 **MARS**
 **SEVERITY**

	STUDY	TERM	ORIGINAL VALUE	NEW VALUE
	MARS	RELATIONSHIP	POSSIBLE REL	POSSIBLY RELATED
	MARS	RELATIONSHIP	UNLIKELY REL	UNLIKELY RELATED
	NEPTUNE	RELATIONSHIP	DEFINITE	DEFINITELY RELATED
	NEPTUNE	RELATIONSHIP	UNRELATED	NOT RELATED
	NEPTUNE	RELATIONSHIP	PROBABLE	PROBABLY RELATED
	PLUTO	RELATIONSHIP	1	NOT RELATED
	PLUTO	RELATIONSHIP	3	POSSIBLY RELATED
	PLUTO	RELATIONSHIP	2	UNLIKELY RELATED
	MARS	SEVERITY	MILD SEVERITY	MILD
	MARS	SEVERITY	MODERATE SEVERITY	MODERATE
	NEPTUNE	SEVERITY	MOD	MODERATE
	NEPTUNE	SEVERITY	SEV	SEVERE
	PLUTO	SEVERITY	1	MILD

EXTRA FEATURE



Resulting data set after modification in the macro call:

```
%update_values (inset =work.ae01
                ,outset=work.ae02
                ,study =
                ,term  =SEVERITY
                ,orgvar=AESEV_ORG
                ,newvar=AESEV_NEW
                );
```

missing

STUDYID	SUBJID	AESEQ	AETERM	AESEV_ORG	AESEV_NEW
MARS	1001	1	NAUSEA	MILD SEVERITY	MILD
MARS	1002	1	DIZZINESS	MODERATE SEVERITY	MODERATE
MARS	1003	1	FEVER	MODERATE SEVERITY	MODERATE
NEPTUNE	1001	1	ABDOMINAL PAIN	MOD	MODERATE
NEPTUNE	1002	1	COUGH	SEV	SEVERE
NEPTUNE	1003	1	VOMITING	MOD	MODERATE
PLUTO	1001	1	COLD	1	MILD
PLUTO	1001	2	NAUSEA	1	MILD
PLUTO	1002	1	DIARRHOEA	1	MILD
SATURN	1001	1	LOSS OF APPETITE	SEVERE	SEVERE
SATURN	1001	2	HEADACHE	MILD	MILD
SATURN	1001	3	NAUSEA	MILD	MILD

AEREL + All other variables

CONCLUSION



- ▶ Clear documentation
 - ▶ Complying with the CDISC Controlled Terminology
 - ▶ Getting consistency
 - ▶ Easy to modify
 - ▶ Preventing hardcoding
 - ▶ Time-saving
-
- ▶ Handling one type of variable per study
 - ▶ Handling one type of variable in all studies

More information: **PhUSE paper DH11**

SAS macros can be downloaded from:

<http://ocs-consulting.nl/phuse2016-dh11>

QUESTIONS



WHEN?

WHERE?

WHAT?

WHO?

HOW?

WHY?

**SAS MACRO
UPDATE_VALUES**

**EFFECTIVELY DEAL WITH
HARDCODING**

**CDISC
CONTROLLED TERMINOLOGY
IN CLINICAL STUDIES**