

Metadata-driven Scientific Use File data management

An approach for integrated data edition and dissemination

doi:10.5281/zenodo.832471 | © BY-SA 4.0

ESRA 2017, University of Lisbon | ISEG N AUD4

20th July 2017

Daniel Bela (LifBi/NEPS)

- 1 Motivation and definition**
- 2 Structuring data edition**
- 3 Defining collaboration interfaces**
- 4 Automating procedures**

1 Motivation and definition

2 Structuring data edition

3 Defining collaboration interfaces

4 Automating procedures

Who I am and what I do

Daniel Bela:

- social scientist
- works for the NEPS at LIfBi
- data manager
- responsible for operative implementation of NEPS data edition processes

NEPS:

- German National Educational Panel Study
- six panel cohorts (from Early childhood to Adult Education)
- each cohort surveyed at least once a year
- team of specialists for data edition at the Research Data Center

Who I am and what I do

Daniel Bela:

- social scientist
- works for the NEPS at LIfBi
- data manager
- responsible for operative implementation of NEPS data edition processes

NEPS:

- German National Educational Panel Study
- six panel cohorts (from Early childhood to Adult Education)
- each cohort surveyed at least once a year
- team of specialists for data edition at the Research Data Center

What I want to talk about

traditional data edition scenario in research projects:

- one or two PhD students are confronted with a bunch of data
- they do their best
- they produce a result dataset
- they (and maybe a few others) make analyses and publish papers
- their PI is happy, the project concludes

What I want to talk about

traditional data edition scenario in research projects:

- one or two PhD students are confronted with a bunch of data
- they do their best
- they produce a result dataset
- they (and maybe a few others) make analyses and publish papers
- their PI is happy, the project concludes

What I want to talk about

traditional data edition scenario in research projects:

- one or two PhD students are confronted with a bunch of data
- they do their best
- they produce a result dataset
- they (and maybe a few others) make analyses and publish papers
- their PI is happy, the project concludes

What I want to talk about

traditional data edition scenario in research projects:

- one or two PhD students are confronted with a bunch of data
- they do their best
- they produce a result dataset
- they (and maybe a few others) make analyses and publish papers
- their PI is happy, the project concludes

What I want to talk about

traditional data edition scenario in research projects:

- one or two PhD students are confronted with a bunch of data
- they do their best
- they produce a result dataset
- they (and maybe a few others) make analyses and publish papers
- their PI is happy, the project concludes

What's the issue with this

result of data edition scenario in research projects:

- usable data (👍!)
- data edition is buried in badly documented syntax files
- even if it is not buried, it is hardly readable for others
- the end data is not replicable
- in case an add-on project (or a follow-up panel wave) happens, someone has to start over again

What's the issue with this

result of data edition scenario in research projects:

- usable data (👍!)
- data edition is buried in badly documented syntax files
- even if it is not buried, it is hardly readable for others
- the end data is not replicable
- in case an add-on project (or a follow-up panel wave) happens, someone has to start over again

What's the issue with this

result of data edition scenario in research projects:

- usable data (👍!)
- data edition is buried in badly documented syntax files
- even if it is not buried, it is hardly readable for others
- the end data is not replicable
- in case an add-on project (or a follow-up panel wave) happens, someone has to start over again

What's the issue with this

result of data edition scenario in research projects:

- usable data (👍!)
- data edition is buried in badly documented syntax files
- even if it is not buried, it is hardly readable for others
- the end data is not replicable
- in case an add-on project (or a follow-up panel wave) happens, someone has to start over again

What's the issue with this

result of data edition scenario in research projects:

- usable data (👍!)
- data edition is buried in badly documented syntax files
- even if it is not buried, it is hardly readable for others
- the end data is not replicable
- in case an add-on project (or a follow-up panel wave) happens, someone has to start over again

How can we do better?

NEPS data edition orchestrated as collaborative work by:

- structuring data edition
- defining interfaces for collaboration
- automating procedures
- *by using structured metadata*

structured metadata:

- any human readable information that is also machine-interpretable
- this may be SQL databases
- this may be CSV files
- this may be other tabular information

How can we do better?

NEPS data edition orchestrated as collaborative work by:

- structuring data edition
- defining interfaces for collaboration
- automating procedures

- *by using structured metadata*

structured metadata:

- any human readable information that is also machine-interpretable
- this may be SQL databases
- this may be CSV files
- this may be other tabular information

1 Motivation and definition

2 Structuring data edition

3 Defining collaboration interfaces

4 Automating procedures

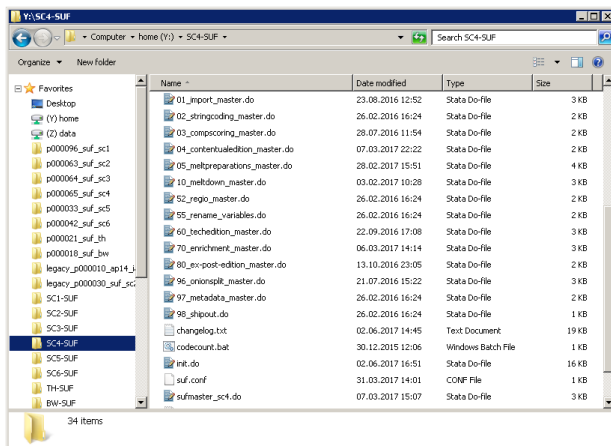
Defining data edition milestones I

this may be something like:

- 1 data import / gathering
- 2 coding of string information
- 3 data compression (i. e. melting down data files)
- 4 variable renaming
- 5 data enrichment
- 6 labeling and exporting data

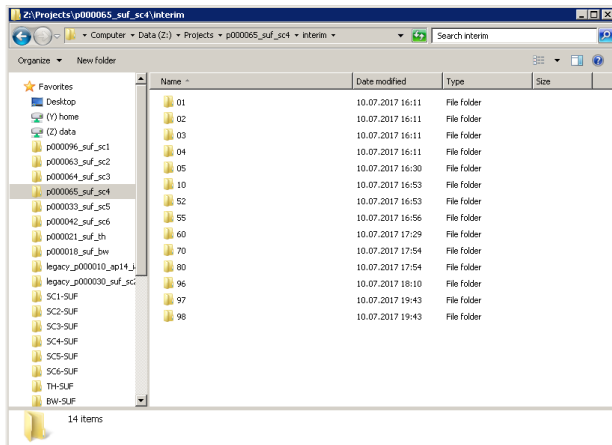
Defining data edition milestones II

the obvious master-slave structure of code files:



Defining data edition milestones III

the not-so-obvious milestone structure of data files:



Defining data edition milestones IV

separation of milestones:

- separate code into steps for each milestone
- also separate (intermediate) datasets for each milestone
- version-control code
- implement *each and any* procedure exclusively in code
- each milestone should be able to be processed standalone (given all defined prerequisites are met)
- clearly define features of datasets output by a milestone (e.g. “all variables have been renamed to the target format”)

What's the gain?

true centralized, but collaborative environment:

- work on different milestones can be done in parallel by different persons
- automated tests checking milestone results can be implemented
- when everything is finished, a central wrapper script (“master file”) can execute the whole process

- 1 Motivation and definition
- 2 Structuring data edition
- 3 Defining collaboration interfaces**
- 4 Automating procedures

Low-level interfaces for collaborators I

issue:

- data edition scripts for complex data have to be complex
- some collaborators won't want to deal with this
- other collaborators won't want to bow to software / platform choices
- but collaboration is necessary to handle complex data edition projects

Low-level interfaces for collaborators II

solution:

wherever collaboration with non-expert collaborators is necessary...

- ...define common human readable (table-based?) interface formats
- ...write a short documentation about these formats
- ...collaborators only need to deal with these tables
- ...resulting tables can be automatically interpreted, and results written to the datasets

Low-level interfaces for collaborators III

example:

sourcedir	sourcefile	targetdir	target	waveind	operation
A48	A48_T_Erst.dta	A48	pTarget	3	append
A48	A48_T_Erst_BB.dta	A48	pTarget	3	append
A48	A48_T_Erst_IndNV.dta	A48	pTarget	3	append
A48	A48_T_Panel.dta	A48	pTarget	3	append
A48	A48_T_Panel_BB.dta	A48	pTarget	3	append
A48	A48_T_Panel_IndNV.dta	A48	pTarget	3	append
A48	A48_T_Statusupdate.dta	A48	pTarget	3	merge 1:1 ID_t wave

- 1 Motivation and definition
- 2 Structuring data edition
- 3 Defining collaboration interfaces
- 4 Automating procedures**

Establishing maintainability

complex and / or panel data edition scripts:

- have to work in the long run
- have to be maintained in the long run

how to achieve this:

- *any* information changing over time should be held in interfaces
- scripts are there to automatically interpret interfaces' content
- repeated operations should be encapsulated in functions
- “don't repeat yourself”

Establishing maintainability

complex and / or panel data edition scripts:

- have to work in the long run
- have to be maintained in the long run

how to achieve this:

- *any* information changing over time should be held in interfaces
- scripts are there to automatically interpret interfaces' content
- repeated operations should be encapsulated in functions
- “don't repeat yourself”

Recap

NEPS data edition approach:

- NEPS data edition is semi-automated
- this is made possible by transferring data edition logic to structured metadata
- data edition routines only interpret and execute these metadata
- certain interfaces enable collaboration despite not all collaborators being tech savvy
- this enables NEPS to disseminate more than six Scientific Use Files per year (on average)

broader perspective:

- structured metadata can be used to:
 - automate data edition procedures
 - define interfaces for collaboration
 - separate edition logic from its implementation

Recap

NEPS data edition approach:

- NEPS data edition is semi-automated
- this is made possible by transferring data edition logic to structured metadata
- data edition routines only interpret and execute these metadata
- certain interfaces enable collaboration despite not all collaborators being tech savvy
- this enables NEPS to disseminate more than six Scientific Use Files per year (on average)

broader perspective:

- structured metadata can be used to:
 - automate data edition procedures
 - define interfaces for collaboration
 - separate edition logic from its implementation

Questions?

Please discuss!



Wilhelmsplatz 3
96047 Bamberg, Germany

Phone: +49 951 863-3428
Fax: +49 951 863-3405
daniel.bela@lifbi.de

www.lifbi.de



License

This presentation is available under the following license:

© Creative Commons Attribution Share-Alike 4.0.

This does not apply to the following material used in this presentation:

the Linux Biolinum and Source Code Pro font families,
as well as all used icons from the Font Awesome font
(all licensed under SIL Open Font license 1.1)

the logo of the Leibniz Institute for Educational Trajectories:

