# Structuring Metadata for Reuse
# Building Foundational Metadata

Marion Wittenberg – DANS
Merja Karjalainen – SND

# Contributors

- Alerk Amin – Centerdata
- Johan Fihn – SND
- Larry Hoyle – University of Kansas
- Merja Karjalainen – SND
- Hilde Orten – NSD
- Bodil Stenvig _ DDA
- Joachim Wackerow – GESIS
- Marion Wittenberg – DANS

# STRUCTURING METADATA FOR REUSE
## BUILDING FOUNDATIONAL METADATA

- Work in progress

- DDI Working Paper Series – Longitudinal Best Practice II, No. 2
  - 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.

# Reuse of metadata

- better support for comparability between different studies
- facilitating the management of references to classifications and other resources

- Foundational Metadata – metadata that are agreed as *de facto* standards

# Publishing Metadata using Resource Packages

- Exclusive publication of reusable items outside the context of a study
- All identifiable items can be reused
- Elements that use a scheme are better candidates for reuse.
- Examples:
  - Category
  - Code
  - Concept
  - Question
  - Universe
  - Variable

# Classifications

- Reuse of a combination of Variable, Code and Categories

- Separate presentation of Hilde Orten and Joachim Wackerow
  *The ISCED Standard for Coding of Education and DDI-Lifecycle*

# Customization of Published Items

- the reusable item/element can be referenced by the user to be used in the original form

  or

- it can be customized to fit into the purpose of the user

# STEPS TO CUSTOMIZE A DDI SCHEME

Two different mechanisms for doing this:

- One mechanism applies to Categories, Concepts, Questions, Universes, and Variables
- The other mechanism applies only to Codes, as DDI implements the CodeMap differently than the other maps.

# Customizing a CategoryScheme, ConceptScheme, QuestionScheme, UniverseScheme, or VariableScheme

1. Create a new Scheme of the appropriate type
2. Use a SchemeReference to include the original scheme
3. Use Exclude for items that are removed
4. Add new items to the customized scheme
5. Modify items.  As there is no explicit way to document a modification in DDI, the solution is to exclude the original item, add the customized item, and document the modifications between these two elements.
6. Create Comparison elements to document items that are excluded/added/modified

# Customizing a CodeScheme

1. Create a new CodeScheme of the appropriate type
2. Use a CodeSchemeReference to include the original scheme
3. List the items that are reused without modification in the customized scheme
4. Add new items to the customized scheme
5. Modify items.  As there is no explicit way to document a modification in DDI, the solution is to exclude the original item, add the customized item, and document the modifications between these two elements.
6. Create Comparison elements to document items that are excluded/added/modified

# How to Reference Elements in a Customized Scheme

Customized scheme can contain a mix of unmodified elements from the original scheme, and the new/modified elements.

- Unmodified elements > refer to the ID/URN in the original scheme
- New/modified element > refer to the ID/URN in the customized scheme, with the new agency.

< ddi >

# Example: Use of a QUESTION BANK

- Reuse of individual questions
  - reuse without any customizations
  - re-use with customizations

- Reuse of a battery of questions

# Reuse of individual questions

▸ Without any customizations > reference the QuestionItem directly from his QuestionConstruct

▸ Re-use with customizations >

1. Create a new QuestionScheme
2. Add a new QuestionItem using the same metadata as the original QuestionItem, but add the translation. This new QuestionItem will be a customized copy of the original QuestionItem, but have a new ID.
3. Create Comparison elements to document the new customized QuestionItem

# Reuse a battery of questions

1. Create a new QuestionScheme, a new ControlConstructScheme and a new InterviewerInstructionScheme
2. Use QuestionSchemeReference, ControlConstructSchemeReference and InterviewerInstructionSchemeReference to include the original schemes
3. Use Exclude for any QuestionItems, QuestionConstructs or Instructions that are removed
4. Add any additional QuestionItems, QuestionConstructs or Instructions to the new schemes.
5. Modify the items which should be translated, e.g. QuestionItem or Instruction
   a. Exclude the original QuestionItem or Instruction as described above
   b. Add new QuestionItems or Instructions with the new translations as described in step 4 of the generic model.
   c. Use the Comparison element to document the modifications. This is described in the following step.
6. Create Comparison elements to document the QuestionItems or Instructions that have been modified.

# Limitations in DDI regarding the use of Question Banks (1)

- No support in DDI for referencing Concepts for an entire ControlConstructScheme  (block of questions)
- There is no Include element in QuestionSchemeReference
- There is no Exclude element in CodeSchemeReference
- The definition of a QuestionScheme should change from (QuestionItem | MultipleQuestionItem)+  to (QuestionItem | MultipleQuestionItem)*

# Limitations in DDI regarding the use of Question Banks (2)

- In ItemMap, the SourceItem and TargetItem should be references (including a version)
- It is not possible to include a Comparison element in a StudyUnit.  Ideally, you would also include a QuestionMap in a QuestionScheme
- IncludedCodeReference is currently a string, but it should be something better for referring to a specific item/version.
- It is not possible to "extend"/"modify" elements

# Thank you!
# Any questions?