

## Historization and Versioning of DDI-Lifecycle Metadata Objects

Findings in the STARDAT Project

5rd Annual European DDI Users Group Meeting  
Paris, 03.-04.12.2013

Alexander Mühlbauer  
GESIS – Leibniz Institute for the Social Sciences

## Outline

- 1 STARDAT at a Glance
- 2 Modeling and Implementing DDI-Lifecycle
- 3 Use Case: Historization and Versioning (H&V)
- 4 Use Case: Import a Statistical Data File

## Outline

- 1 STARDAT at a Glance
- 2 Modeling and Implementing DDI-Lifecycle
- 3 Use Case: Historization and Versioning (H&V)
- 4 Use Case: Import a Statistical Data File

## Objectives

- Integrated management system for metadata
- Replacement for production systems DBK, DSDM, CBE
- Multi-language documentation on study and variable level
- Controlled vocabularies (Thesauri)
- Related publications, scales, trends and add. metadata
- Interfaces to portals like ZACAT, Cessda, Data Portal, Sowipor and da|ra
- Support of and high interoperability with standards like DDI-C and DDI-L
- Longterm-preservation with DDI

## Outline

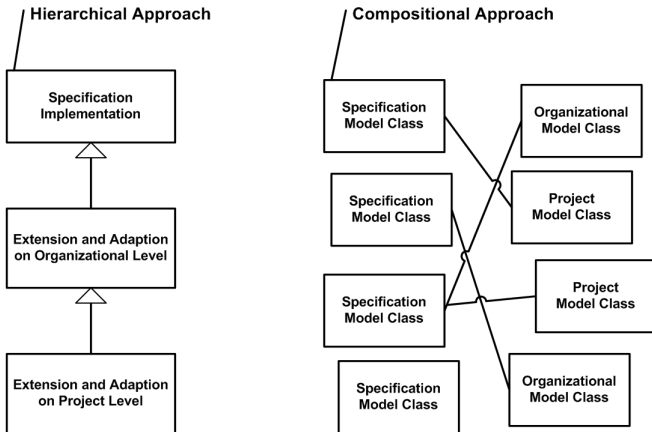
- 1 STARDAT at a Glance
- 2 Modeling and Implementing DDI-Lifecycle
- 3 Use Case: Historization and Versioning (H&V)
- 4 Use Case: Import a Statistical Data File

## Principles for Analysis, Design and Coding

### My Guidelines for Daily Work

- Strict separation of *modeling* and *mapping*
  - avoids specific, not reusable implementations
  - allows use case specific, possibly different mappings
  - makes incompatibilities explicit
  
- Strict separation of *model* and *representation*
  - leads to true abstraction in memory
  - is the very necessary base for true interchangeability

## Towards a GESIS DDI Architecture



## Towards a GESIS DDI Architecture

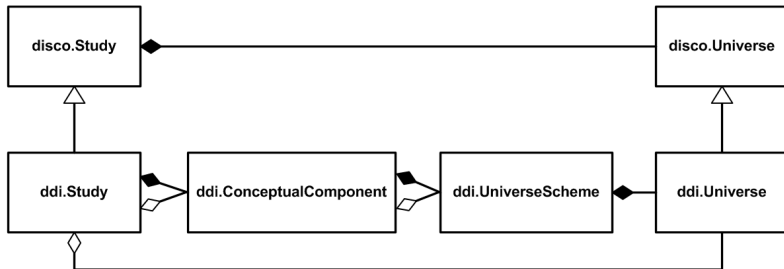
### Comparison of the Two Approaches

- Hierarchy (Inheritance)
  - Technical infrastructure on organizational level (+)
  - Support of services out of the box (+)
  - Lower-level components must be true extensions (-)
  - Explicit definition of requirements for extensibility (-)
- Composite (Mapping)
  - Independence and flexibility in modeling (+)
  - Making incompatibilities explicit (+)
  - Re-implementation of components (-)



## Towards a GESIS DDI Architecture

Implementation Challenge: Loss of Maintainables



## Coping with DDI Schema Change

### My Guidelines for Daily Work

- **Accept it!!!**
- Extract it!
- Restrict it!
- Anticipate it!
- **Contribute to abstract it!**

## Coping with DDI Schema Change

Example: Label – Extract and Restrict It!

- DDI 3.1: Object may have **one** multi-lingual label
- DDI 3.2: Object may have **many** multi-lingual labels

⇒ Is it a critical requirement?

## Coping with DDI Schema Change

Example: Note – Anticipate It!

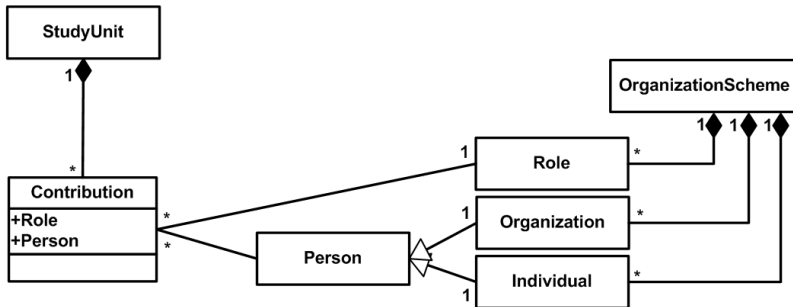
- DDI 3.1: Note is an Identifiable
- DDI 3.2: Note is not an Identifiable any more

⇒ Resolve DDI-L inheritance hierarchy on technical level!

- Only distinction between Identifiable and Non-Identifiable
- All types with domain responsibility are Identifiable

## Coping with DDI Schema Change

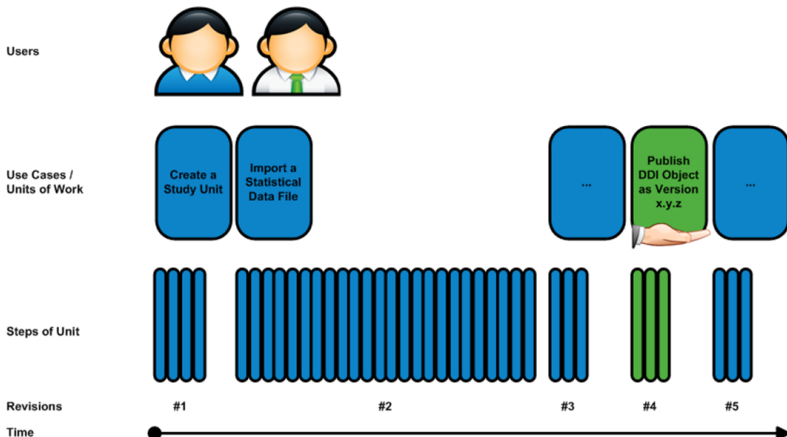
Example: Citation – Abstract It!



## Outline

- 1 STARDAT at a Glance
- 2 Modeling and Implementing DDI-Lifecycle
- 3 Use Case: Historization and Versioning (H&V)
- 4 Use Case: Import a Statistical Data File

## Use Case: Historization and Versioning



## Definition: *Historization*

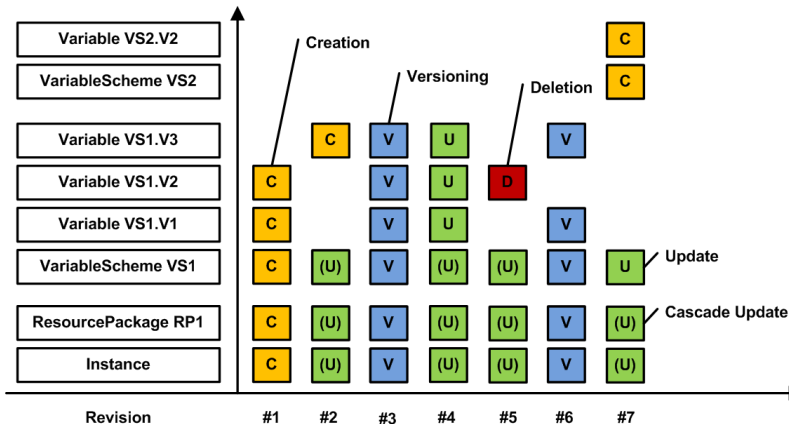
- is the technical process
- to keep track of change
  - which metadata objects changed
  - when
  - why
  - by whom
- within a transaction
- on attribute and relationship level;
- builds the foundation of *Versioning*.



## Definition: *Versioning*

- is the business process
- to flag a given metadata object
- at a given revision with a version number
- according to the agency's versioning policy
- if possible with recommendation or automatically.

## Use Case: Historization and Versioning



## H&V Use Case: Revision View

- Show given revision, range of revisions or all revisions with included metadata objects

The screenshot shows a web browser at the URL <https://apps.codenomics.de/stardat-snoopy/revision/list>. The page title is "STARDAT Snoopy Prototype: Historization and Versioning of DDI-Lifecycle Metadata Objects". The navigation bar includes "Database", "Revisions" (highlighted), "Resources", and "Resource Revision Diff".

The main content area displays two revision entries:

**Revision 7 by Catharina at 03.12.2013 10:25:00: Create VariableScheme VS2 with Variable VS2.V2 and add a reference of VS2.V2 to VS1**

Metadata Object Type	Identifier	Representations			
Instance	3971cd1c-b838-4bbe-aa80-e46deaa6207b	<a href="#">HTML</a>	<a href="#">DDI 3.1</a>	<a href="#">DDI 3.2</a>	<a href="#">PDF</a>
Variable	48201f56-da45-472c-8341-9eed0a916536	<a href="#">HTML</a>	<a href="#">DDI 3.1</a>	<a href="#">DDI 3.2</a>	<a href="#">PDF</a>
ResourcePackage	e3de3bfd-6b18-46e3-98c5-5ebb272b8add	<a href="#">HTML</a>	<a href="#">DDI 3.1</a>	<a href="#">DDI 3.2</a>	<a href="#">PDF</a>
VariableScheme	07816a2b-511a-406d-9e57-e82b6c141920	<a href="#">HTML</a>	<a href="#">DDI 3.1</a>	<a href="#">DDI 3.2</a>	<a href="#">PDF</a>
VariableScheme	77f6e18b-7fc2-4995-900d-1f2ec91b2789	<a href="#">HTML</a>	<a href="#">DDI 3.1</a>	<a href="#">DDI 3.2</a>	<a href="#">PDF</a>

**Revision 6 by Monika at 02.12.2013 19:35:00: Flag VariableScheme VS1 at current revision as version 2.0.0, the variables as 1.1.0**

Metadata Object Type	Identifier	Representations			
Instance	3971cd1c-b838-4bbe-aa80-e46deaa6207b	<a href="#">HTML</a>	<a href="#">DDI 3.1</a>	<a href="#">DDI 3.2</a>	<a href="#">PDF</a>

## H&V Use Case: Resource View

- Show given resource in current state and older revisions

The screenshot shows a web browser window with the URL <https://apps.codenomics.de/stardat-snoopy/resource/list>. The application header includes the 'gesis' logo and a 'Contact' link. The main title is 'STARDAT Snoopy Prototype: Historization and Versioning of DDI-Lifecycle Metadata Objects'. A navigation bar contains links for 'Database', 'Revisions', 'Resources' (which is highlighted), and 'Resource Revision Diff'. Below this, a table displays the history of revisions for a specific variable.

Variable 0d9753ac-e194-40fc-920b-890858f3ade							
#	Revision Date	Comment	User	Representations			
1	01.12.2013 08:10:00	Create Instance I1, ResourcePackage RP1, VariableScheme VS1 and Variables VS1.V1 and VS1.V2	Monika	<a href="#">HTML</a>	<a href="#">DDI 3.1</a>	<a href="#">DDI 3.2</a>	<a href="#">PDF</a>
3	02.12.2013 09:40:00	Flag VariableScheme VS1 at current revision as version 1.0.0	Catharina	<a href="#">HTML</a>	<a href="#">DDI 3.1</a>	<a href="#">DDI 3.2</a>	<a href="#">PDF</a>
4	02.12.2013 11:00:00	Add german labels for variables of VariableScheme VS1	Wolfgang	<a href="#">HTML</a>	<a href="#">DDI 3.1</a>	<a href="#">DDI 3.2</a>	<a href="#">PDF</a>
6	02.12.2013 19:35:00	Flag VariableScheme VS1 at current revision as version 2.0.0, the variables as 1.1.0	Monika	<a href="#">HTML</a>	<a href="#">DDI 3.1</a>	<a href="#">DDI 3.2</a>	<a href="#">PDF</a>

Variable 12a229d0-ef47-4e84-adfa-667dac634c62							
#	Revision Date	Comment	User	Representations			
2	02.12.2013 09:35:00	Create Variable VS1 V3	Wolfgang	<a href="#">HTML</a>	<a href="#">DDI 3.1</a>	<a href="#">DDI 3.2</a>	<a href="#">PDF</a>

## H&V Use Case: Resource Diff View

- Compare given resource between two selected revisions

The screenshot shows a web browser at the URL <https://apps.codenomics.de/stardat-snoopy/resource/list/diff>. The page title is "STARDAT Snoopy Prototype: Historization and Versioning of DDI-Lifecycle Metadata Objects". The navigation bar includes links for "Database", "Revisions", "Resources", and "Resource Revision Diff" (which is highlighted in orange). The main content area displays a table for Variable 0d9753ac-e194-40fc-920b-890858fc3ade. The table has columns for #, Revision Date, Comment, User, and Representations. The table shows four revisions, with revision 4 selected. A "Compare" button is visible at the bottom of the table. Below the table, another section for Variable 12a229d0-ef47-4e84-adfa-667dac634c62 is partially visible.

#	Revision Date	Comment	User	Representations
1	01.12.2013 08:10:00	Create Instance I1, ResourcePackage RP1, VariableScheme VS1 and Variables VS1.V1 and VS1.V2	Monika	<a href="#">HTML</a> <a href="#">DDI 3.1</a> <a href="#">DDI 3.2</a> <a href="#">PDF</a>
3	02.12.2013 09:40:00	Flag VariableScheme VS1 at current revision as version 1.0.0	Catharina	<a href="#">HTML</a> <a href="#">DDI 3.1</a> <a href="#">DDI 3.2</a> <a href="#">PDF</a>
4	02.12.2013 11:00:00	Add german labels for variables of VariableScheme VS1	Wolfgang	<a href="#">HTML</a> <a href="#">DDI 3.1</a> <a href="#">DDI 3.2</a> <a href="#">PDF</a>
6	02.12.2013 19:35:00	Flag VariableScheme VS1 at current revision as version 2.0.0, the variables as 1.1.0	Monika	<a href="#">HTML</a> <a href="#">DDI 3.1</a> <a href="#">DDI 3.2</a> <a href="#">PDF</a>

Compare

Variable 12a229d0-ef47-4e84-adfa-667dac634c62

#	Revision Date	Comment	User	Representations
2	02.12.2013 09:35:00	Create Variable VS1.V2	Wolfgang	<a href="#">HTML</a> <a href="#">DDI 3.1</a> <a href="#">DDI 3.2</a> <a href="#">PDF</a>

## H&V Use Case: DDI 3.1 Representation

```
-<ddi:DDIInstance agency="de.gesis" id="3971cd1c-b838-4bbe-aa80-e46deaa6207b" version="7" isPublished="false">
  <r:UserID type="11179-IRDI">de.gesis:3971cd1c-b838-4bbe-aa80-e46deaa6207b:7</r:UserID>
- <g:ResourcePackage agency="de.gesis" id="e3de3bfd-6b18-46e3-98c5-5ebb272b8add" version="7"
  isPublished="false">
  <r:UserID type="11179-IRDI">de.gesis:e3de3bfd-6b18-46e3-98c5-5ebb272b8add:7</r:UserID>
- <l:VariableScheme id="77f6e18b-7fc2-4995-900d-1f2ec91b2789" agency="de.gesis" version="7"
  isPublished="false">
  <r:UserID type="11179-IRDI">de.gesis:77f6e18b-7fc2-4995-900d-1f2ec91b2789:7</r:UserID>
- <l:Variable id="0d9753ac-e194-40fc-920b-890858fc3ade" version="1.1.0">
  <r:UserID type="11179-IRDI">de.gesis:0d9753ac-e194-40fc-920b-890858fc3ade:6</r:UserID>
  <l:VariableName>VS1.V1</l:VariableName>
  <r:Label xml:lang="en">Confidence in parliament</r:Label>
  <r:Label xml:lang="de">Vertrauen in Parlament</r:Label>
  </l:Variable>
- <l:VariableReference>
  <r:ID>48201f56-da45-472c-8341-9eed0a916536</r:ID>
  <r:IdentifyingAgency>de.gesis</r:IdentifyingAgency>
  <r:Version>7</r:Version>
  </l:VariableReference>
- <l:Variable id="12a229d0-ef47-4e84-adfa-667dac634c62" version="1.1.0">
  <r:UserID type="11179-IRDI">de.gesis:12a229d0-ef47-4e84-adfa-667dac634c62:6</r:UserID>
  <l:VariableName>VS1.V3</l:VariableName>
  <r:Label xml:lang="en">Confidence in schools and educational system</r:Label>
  <r:Label xml:lang="de">Vertrauen in Schulen und Bildungssystem</r:Label>
  </l:Variable>
  </l:VariableScheme>
- <l:VariableScheme id="07816a2b-511a-406d-9e57-e82b6c141920" agency="de.gesis" version="7"
  isPublished="false">
  <r:UserID type="11179-IRDI">de.gesis:07816a2b-511a-406d-9e57-e82b6c141920:7</r:UserID>
  <l:Variable id="48201f56-da45-472c-8341-9eed0a916536" version="7">
```

## H&V Use Case: Implementation Details

- Not an implementation image of the specification but close as possible to read and write DDI-L
- Java objects bundled as Maven Module
- Java Persistence API and Hibernate Envers annotations as persistence abstraction
- Hibernate as persistence provider

### Note!

- Revision number is a global identifier of repository state.
- Version number is not a property of a metadata class.

## Object Identification

- $\{\text{agency}\}:\{\text{identifier}\}:(\{\text{revision}\})$ , e.g.  
de.gesis:3971cd1c-b838-4bbe-aa80-e46deaa6207b:7
- $\{\text{agency}\}:\{\text{identifier}\}:\{\text{version}\}$ , e.g.  
de.gesis:3971cd1c-b838-4bbe-aa80-e46deaa6207b:2.0.0
- Identifier unique within the maintaining agency
- Revision as an ascending Integer
- Version as composition of ascending Integers separated by dots
- Separation of contents/concepts and identification



## H&V Use Case: Base Class Identifiable With Hibernate Envers Annotations

```
@Entity
@Audited
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
public class Identifiable
{
    @Id
    private String id;

    @Column
    private String agency;

    @Column
    private long lastModified;

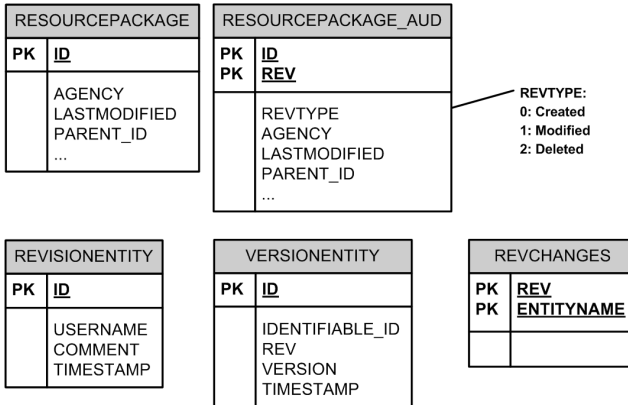
    @ManyToOne
    private Identifiable parent;

    // ...

    @PrePersist
    protected void cascadeLastModified()
    {
        lastModified = RevisionContext.getInstance().getTimestamp();
        if (this.parent != null)
            this.parent.cascadeLastModified();
    }
}
```

## H&V Use Case: Entity Relationship Diagram

Generated by Hibernate Envers



## H&V Use Case: Prototype

- Read-only mode for inspecting the small demo use case
- See database tables and contents
- <https://apps.codenomics.de/stardat-snoopy>

The screenshot shows a web browser window with the URL <https://apps.codenomics.de/stardat-snoopy/database>. The page header includes the 'gesis' logo and a 'Contact' link. The main title is 'STARDAT Snoopy Prototype: Historization and Versioning of DDI-Lifecycle Metadata Objects'. Below the title is a navigation bar with tabs: 'Database' (selected), 'Revisions', 'Resources', and 'Resource Revision Diff'. The 'Database' tab displays a table titled 'Table: REVISIONENTITY' with the following data:

ID	TIMESTAMP	COMMENT	USERNAME
1	1385881800000	Create Instance I1, ResourcePackage RP1, VariableScheme VS1 and Variables VS1.V1 and VS1.V2	Monika
2	1385973300000	Create Variable VS1.V3	Wolfgang
3	1385973600000	Flag VariableScheme VS1 at current revision as version 1.0.0	Catharina
4	1385978400000	Add german labels for variables of VariableScheme VS1	Wolfgang
5	1386000000000	Delete Variable VS1.V2	Olav

## Outline

- 1 STARDAT at a Glance
- 2 Modeling and Implementing DDI-Lifecycle
- 3 Use Case: Historization and Versioning (H&V)
- 4 Use Case: Import a Statistical Data File

## Use Case: Import a Statistical Data File

### Present Workflow and Implementation

- Statistical data file as content reference
- Only support of SPSS
- Import and iterative update of variable information
- Entities: Study, Variable, VariableValue
- All assoziations are compositions (no reuse)

## Use Case: Import a Statistical Data File

### Possible Future Workflow and Implementation

- Present workflow must be supported
- Also Documentation as content reference
- Entities: StudyUnit|ResourcePackage, Variable, Code, Category, LogicalRecord, PhysicalStructure, RecordLayout, DataItem and necessary schemes
- Some associations are compositions, some are aggregations (reuse by reference)

## Use Case: Import a Statistical Data File

### Comparison

- More complex domain model (-/+)
- More comprehensively complying requirements (+)
- Different thinking about imports and iterative update required by reuse possibilities
- Very intuitive user interface needed to not get lost

## Use Case: Import a Statistical Data File

### Staged Reuse of Imported Variable Description

- Import all information without any (!) reuse (stage 1)
- Propose use of / Allow search for reusable items
  - within the dataset,  
e.g. code and category schemes (stage 2)
  - within project scope,  
e.g. variables (stage 3)
  - within agency scope,  
e.g. resource packages (stage 4)
  - beyond agency scope (stage 5)



## Summary and Conclusion

- Object-relational modeling of DDI-Lifecycle is possible and may help to find suitable abstractions for future DDI versions.
- Hibernate Envers with JPA offers a nice solution to implement historization and versioning of DDI-Lifecycle metadata objects.
- Not only software tools, but (even simple) workflows and processes change fundamentally with the usage of DDI-Lifecycle.

Thank you!  
Any questions?



[alexander.muehlbauer@gesis.org](mailto:alexander.muehlbauer@gesis.org)  
GESIS – Leibniz Institute for the Social Sciences