The DASISH Questionnaire Design Documentation Tool: Updates on the development of the tool

Benjamin Beuster, Stig Norland, Hilde Orten
NSD – Norwegian Social Science Data Services

EDDI 2015 - 7th Annual European DDI User Conference, Copenhagen, December 2 – 3 2015

Outline

- The DASISH Questionnaire Design Documentation Tool (QDDT) and the Question Variable Database (QVDB), update

- The Questionnaire Design Documentation Tool (QDDT)
  - New conceptual model (extract of conceptual model for the QVDB)
  - New technical approach

The QDDT and QVDB

- Tools for surveys under development by NSD.
- Work on the tools started under the Data Service Infrastructure for the Social Sciences and Humanities (DASISH) project. It now continues under Synergies for Europe's Research Infrastructure in the Social Sciences (SERISS).
- Our work on the tools was presented at EDDI 2013 and 2014
- Material developed under DASISH is available on GitHub

QDDT and QVDB – recap of aims

- Assist large-scale survey projects in processes related to questionnaire development, data and metadata production, archiving and dissemination

- Internal users:
  - QDDT: Document the questionnaire design process and develop questionnaires
  - QVDB: Develop variables, standards and data products

- External users:
  - QDDT: Understand how each item and concept was developed (QDDT)
  - QVDB: Find and reuse questions and variables (QVDB)

- Reusable model and code

- Interoperability with DASISH Translation Management Tool (TMT) and other tools, using DDI

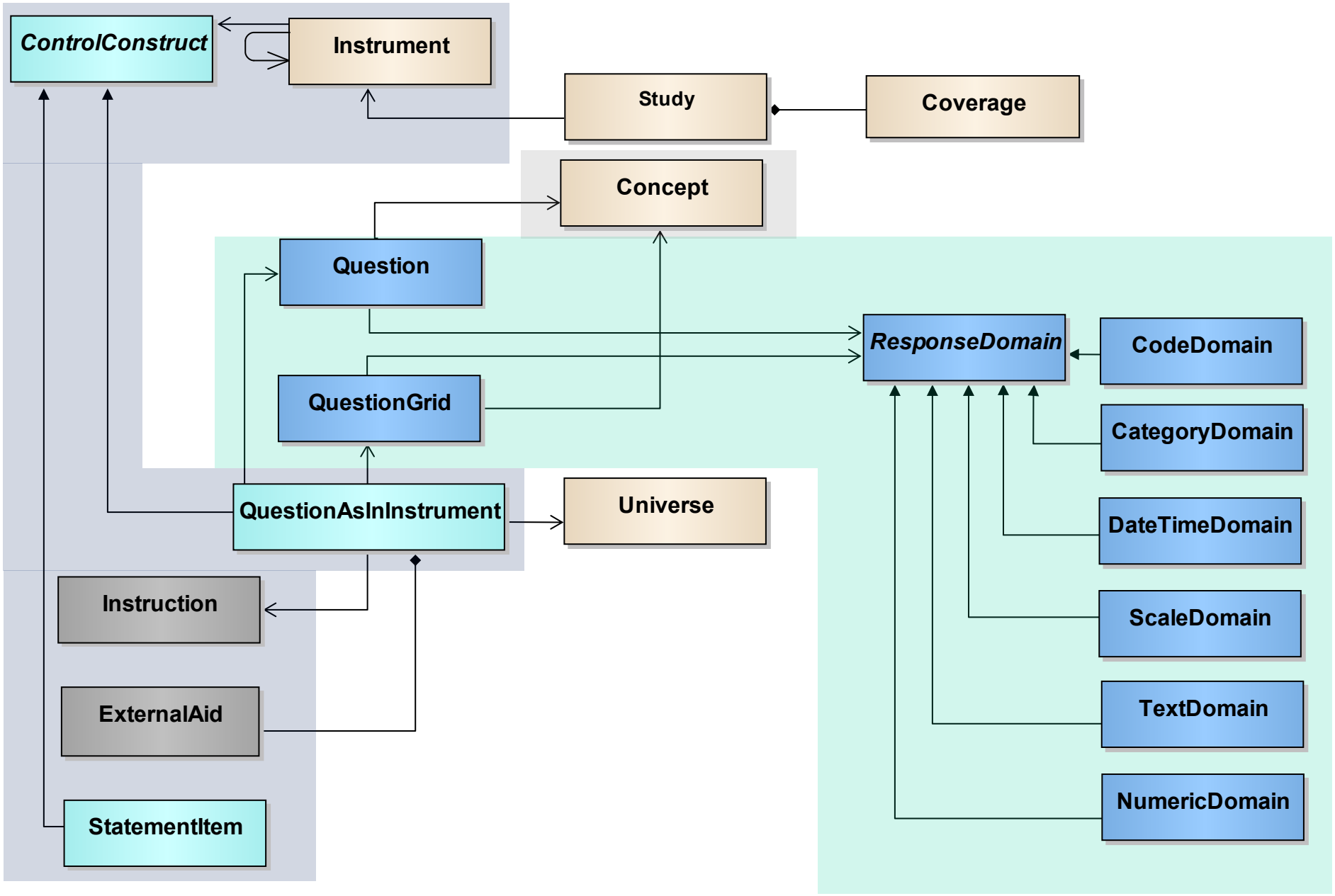
QDDT and QVDB – What was achieved while DASISH?

- A common metadata model from the three DASISH tools, QDDT, QVDB and the TMT
- Specifications for the QDDT and the QVDB
- High level conceptual model for the QVDB
- A prototype of the QDDT

Further developments: What will be kept and what is new?

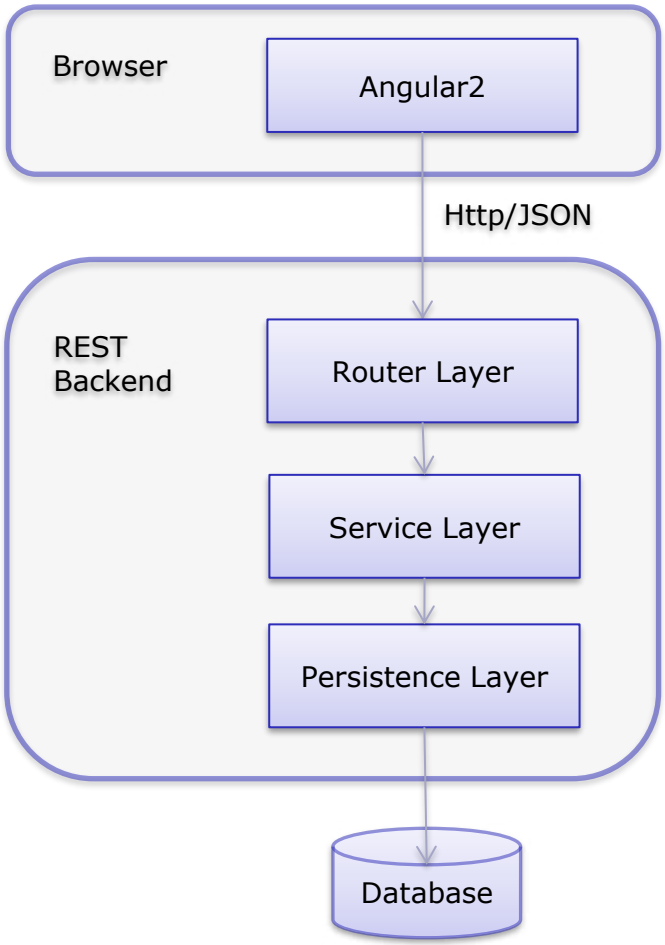
- The common metadata model and the specifications for the QDDT and the QVDB will be kept
- Conceptual model for the QVDB has been detailed. An extract of this works as a model for the QDDT.
- A new prototype of the QDDT based using new technologies is currently under development
- The development of the QVDB builds on the QDDT. One or two tools remains to be decided.

Conceptual model QDDT (extract of model for QVDB)



QDDT Architecture

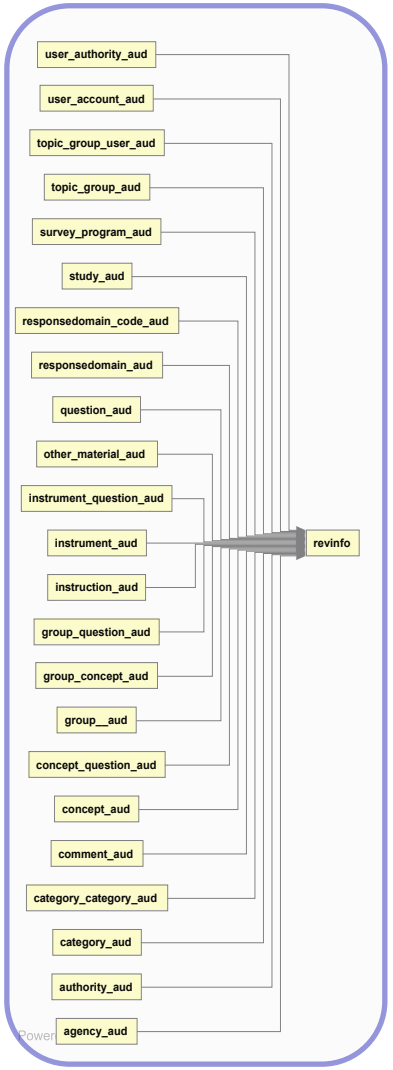
- Angular 2 + Typescript
  - No Javascript in dev tools, only in browser
  - Typescript can transpile to ES3
  - Supports «evergreen» browsers (Chrome, FireFox, Opera, Safari, and IE10/11)
- Spring MVC API
  - RESTful API
  - JSON for free
  - no need for a SOAP service stack
- Spring security (OAuth2)
  - Tokenbased
  - No state for requests
- Persistence Layer implemented with Hibernate + Envers
  - Revisions
  - Database agnostic
  - Since qddt is open source, we chose Postgres



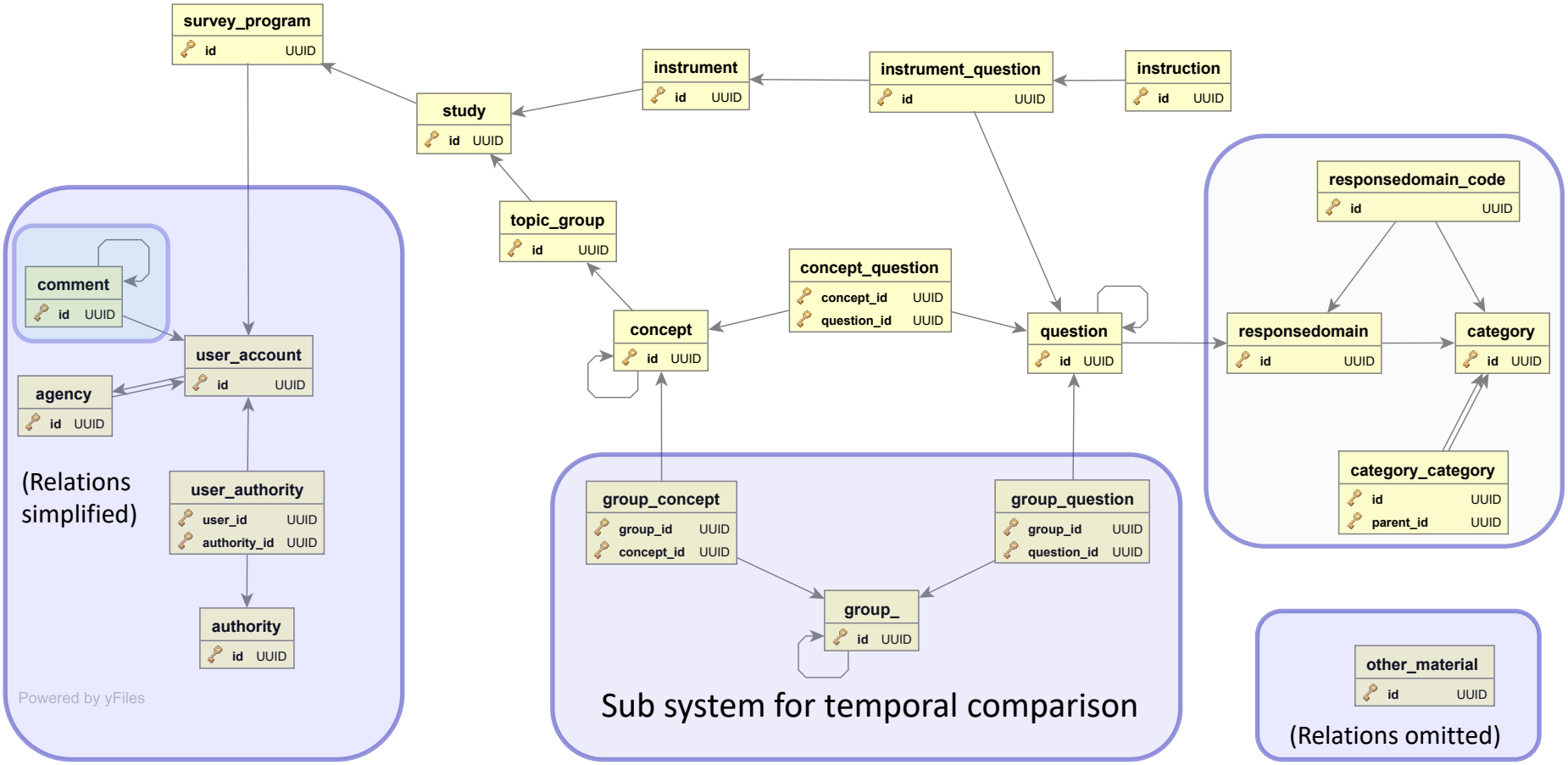
Versioning

- Provide historical versioning of entity data
- Each transaction relates to one global revision number
- Can be used to identify groups of changes (much like a changeset in source control)
- Any changeset can be labeled (business version)
- Find revision by date or label name
- Controlled vocabulary controls business versioning
- Envers
  - Implemented by anotations in domian classes and inclusion of `hibernate-envers-jar` in classpath

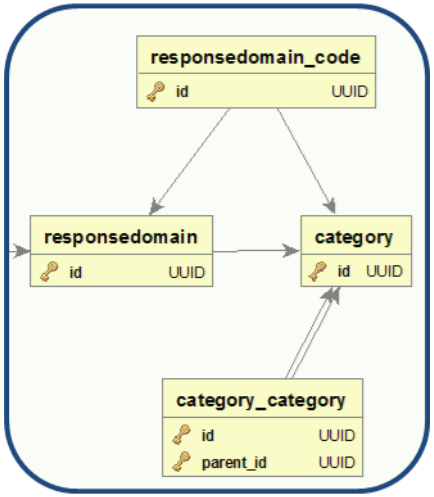
```
32      *
33      * @author Stig Norland
34      * @author Dag Østgulen Heradstveit
35      */
36     @Audited
37     @Entity
38     @Table(name = "CATEGORY")
39     public class Category extends AbstractEntityAudit {
```



Data model


survey_program
id UUID
instrument
id UUID
instrument_question
id UUID
instruction
id UUID
study
id UUID
topic_group
id UUID
concept_question
concept_id UUID
question_id UUID
concept
id UUID
question
id UUID
responsedomain_code
id UUID
responsedomain
id UUID
category
id UUID
category_category
id UUID
parent_id UUID
comment
id UUID
user_account
id UUID
agency
id UUID
(Relations simplified)
user_authority
user_id UUID
authority_id UUID
authority
id UUID
Powered by yFiles
group_concept
group_id UUID
concept_id UUID
group_question
group_id UUID
question_id UUID
group_
id UUID
Sub system for temporal comparison
other_material
id UUID
(Relations omitted)

Category group structure



Category Group

«managed scale representation»

«Scale dimension»

RANGE

Category

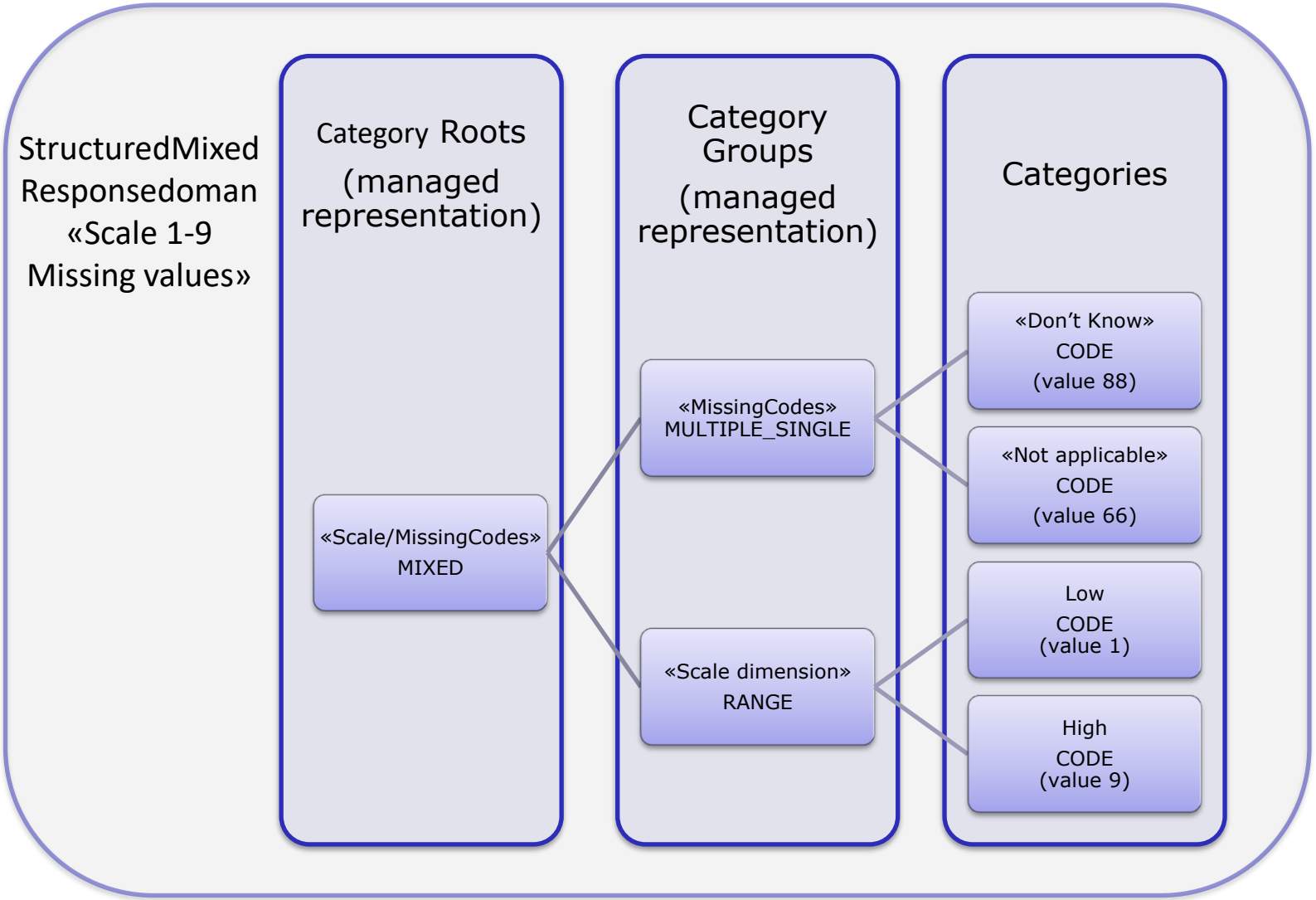
Low

CODE

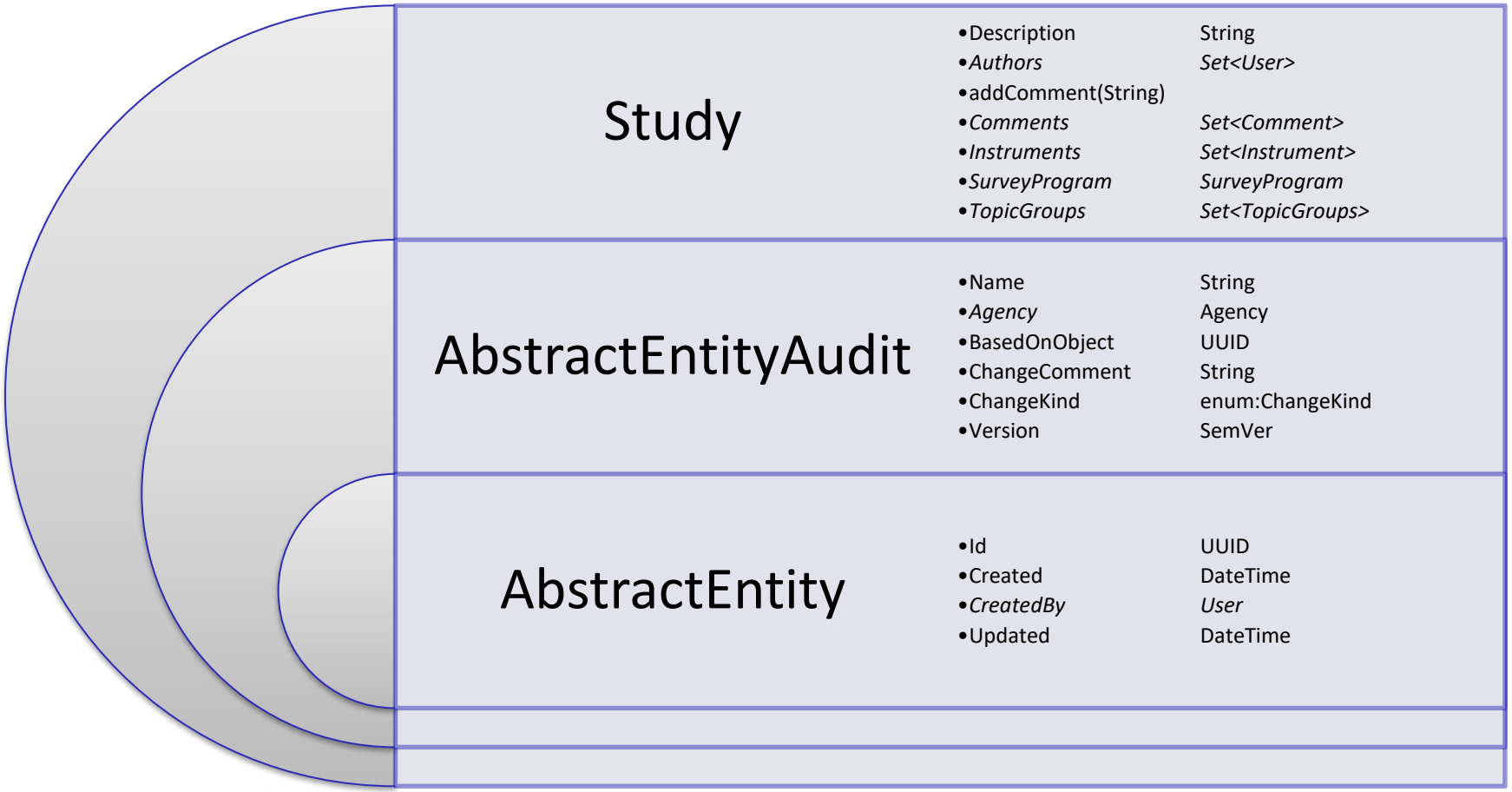
High

CODE

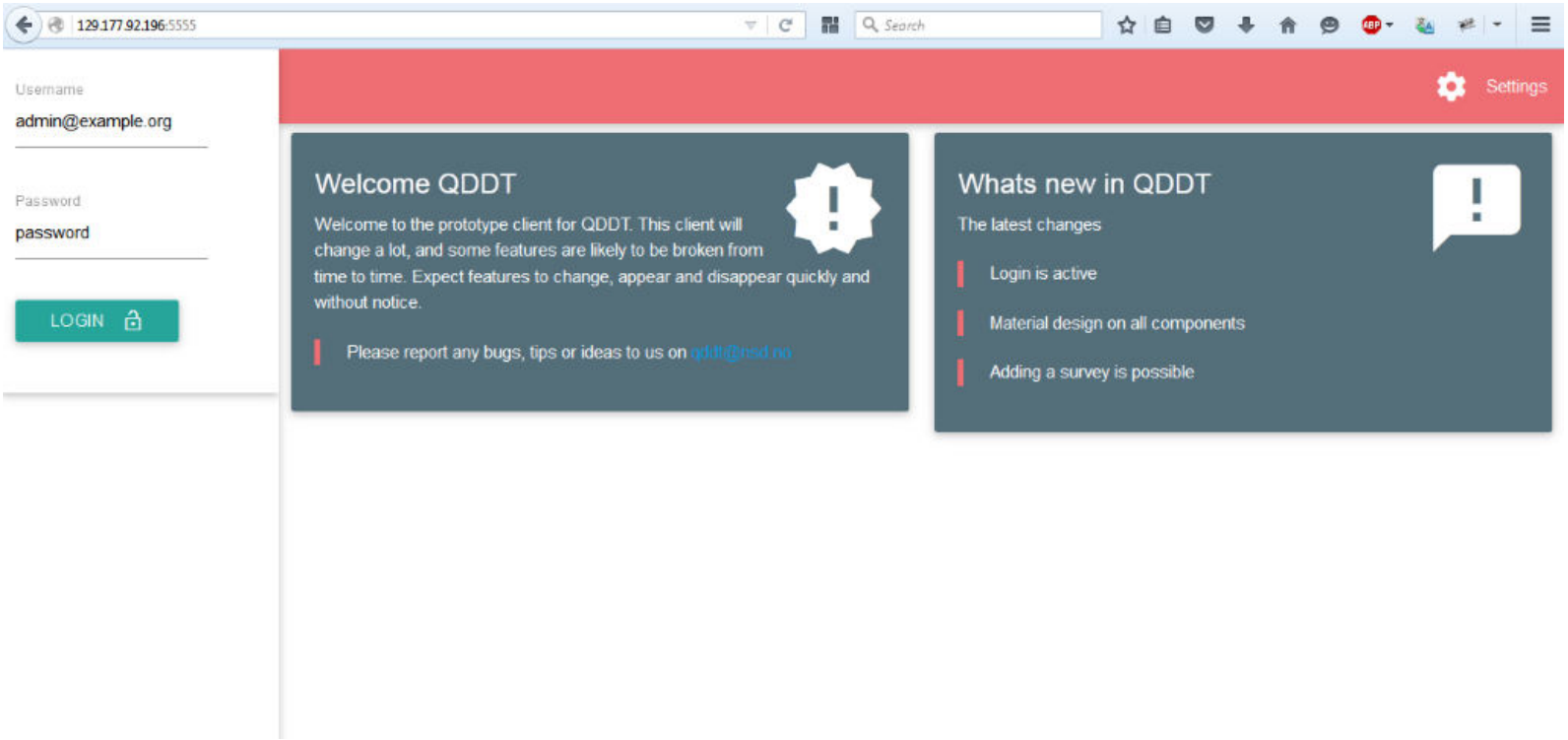
Responsdomain root category structure



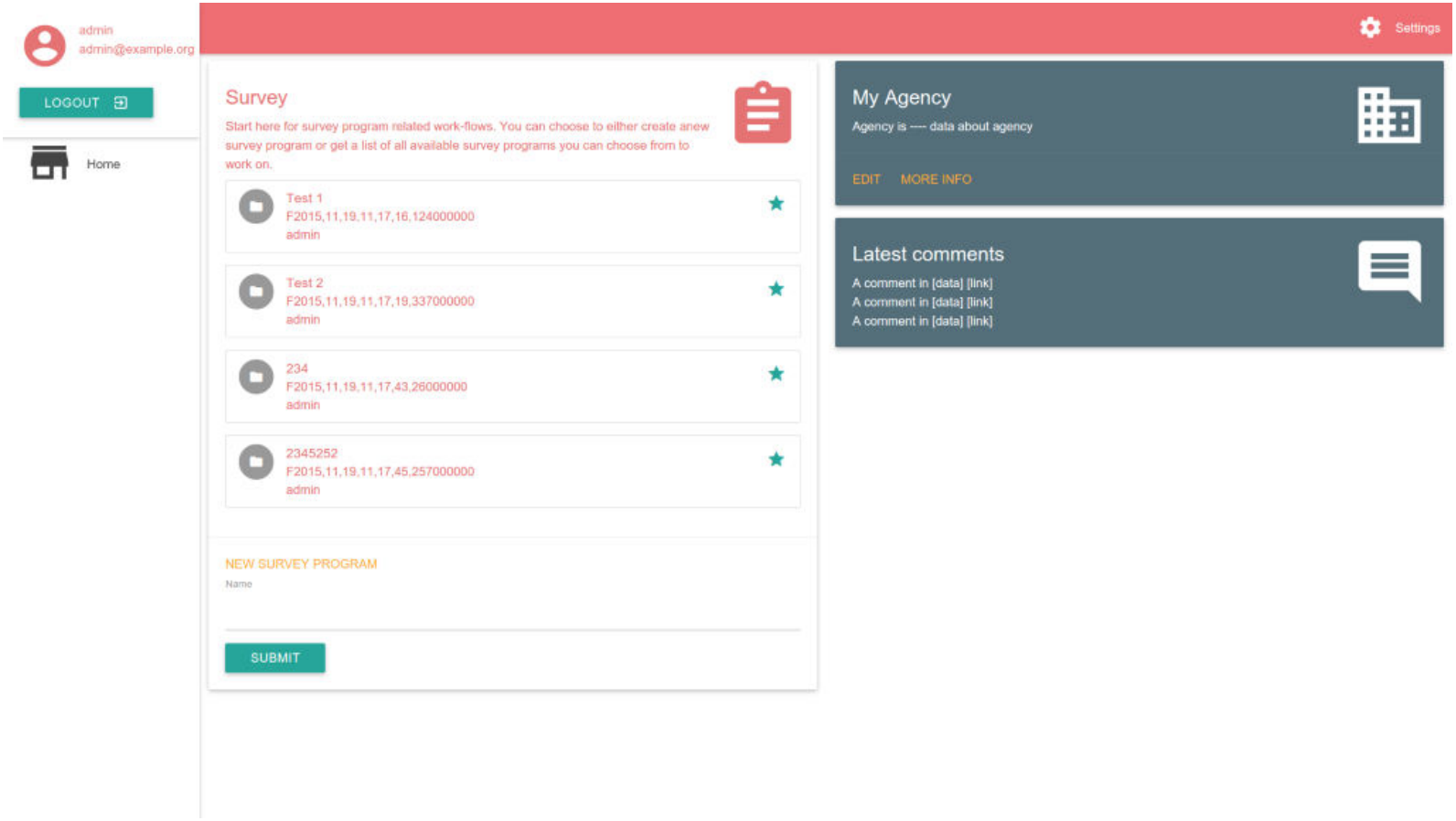
Design of a domain class



Login screen



Survey screen



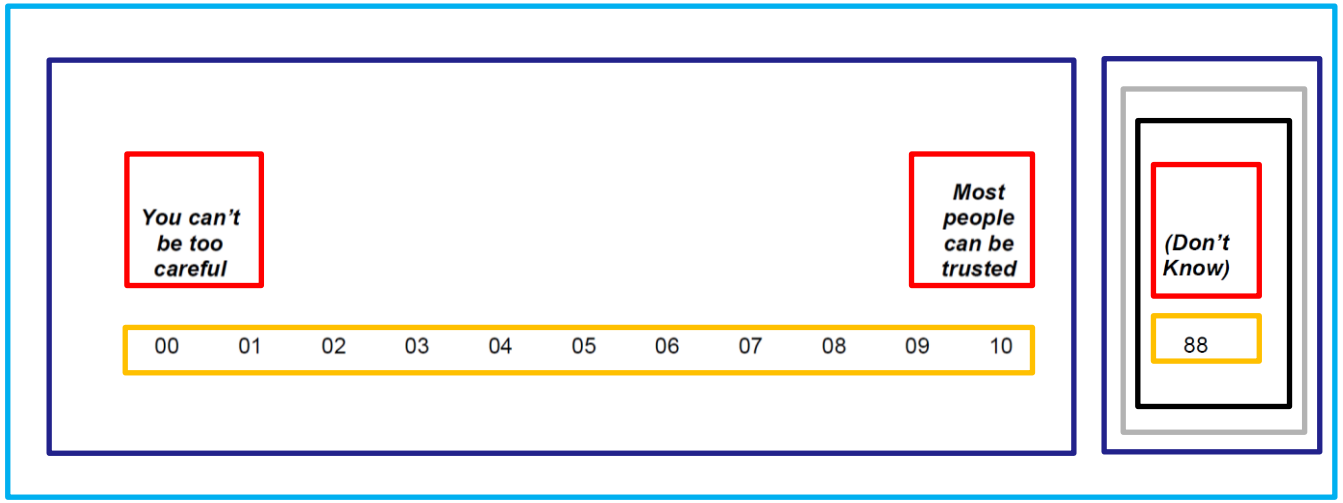
Acknowledgements

Special thanks to Joachim Wackerow (consultant), Dag Østgulen Heradstveit (NSD) and Knut Kalgraff Skjåk (NSD)

Thank you for your attention!

surveytools@nsd.no

DDI StructuredMixedResponseDomain, example



Not versionable	Versionable	
		Categories, Anchor
		Values; NumberRange for scale (high, low, min, max)
		Code
		CodeList
		ManagedRepresentation (Scale, MissingValues)
		StructuredMixedResponseDomain