Catalogue of Quality Problems in Data, Data Models and Data Transformations

Requirements specification for the quality management, quality analysis techniques, improvement measures as well as quality management processes of data sets for objects of the material culture

Developed as part of the BMBF-funded project KONDA

KONDA - Continuous quality management of dynamic research data on objects of material culture using the LIDO standard

Authors	Arno Kesper, Markus Matoni, Julia Rössel, Michelle Weidling, Viola Wenz
Date	2020-09-30
Version	1.0
Status	published
Release	2020-10-14

Bundesministerium für Bildung und Forschung
Philipps Universität Marburg
GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN
NIEDERSÄCHSISCHE STAATS- UND UNIVERSITÄTSBIBLIOTHEK GÖTTINGEN | SUB

Table of Contents

1. Introduction

This catalogue describes quality problems in research data that are focussed on material cultural objects, corresponding data models and data transformations. It was created within the KONDA research project. This catalogue aims to elicit requirements for a generic quality management process for research data through the exchange with the community. The catalogue is the result of qualitative interviews and a workshop with experts on cultural heritage data. Within the catalogue, each quality problem is described through a structured profile, including aspects such as the affected quality dimensions, examples, causes and ideas on improvement.

1.1. KONDA Project Description and Affiliates

KONDA is an interdisciplinary joint research project at the Universities of Göttigen and Marburg, funded by the Federal Ministry of Education and Research (BMBF) for three years, from 2019 to 2022. The acronym "KONDA" stands for continuous quality management of dynamic research data on objects of material culture using the LIDO standard.

The project intends to develop a systematic quality assurance of structured research data on objects of material culture, a desideratum for research in the humanities and cultural sciences. The approach of a continuous quality management process (QM process) differentiating according to data, data models and data transformations over the entire lifecycle of data is groundbreaking and pioneering. Therefore, a generic QM process for dynamic, partly uncertain research data will be developed. This process is applied to the internationally accepted harvesting format Lightweight Information Describing Objects (LIDO) for objects of material culture. It is then converted into specified curatorial criteria for data generation and curation of art historical research data describing various genres of material objects that are collected e.g. in museums or university collections.
The QM Process is to be tested on selected databases e.g. of the Deutsches Dokumentationszentrum für Kunstgeschichte – Bildarchiv Foto Marburg and the University Collections of the University of Göttingen. Resulting QM processes are documented in manuals and will be made available to the professional community.

Affiliates

- Philipps-Universität Marburg / Deutsches Dokumentationszentrum für Kunstgeschichte – Bildarchiv Foto Marburg (DDK)[1]
- Göttingen State and University Library[2]
- Philipps-Universität Marburg / Department of Mathematics and Computer Science[3]

1.2. Approach

While KONDA as a project aims to develop a quality management process which focuses on research data, especially data of material cultural objects, this process has to be grounded in specific requirements. As experience has made evident the chance of changes being accepted by a group of

[1] https://www.uni-marburg.de/de/fotomarburg
[2] https://www.sub.uni-goettingen.de/
[3] https://www.uni-marburg.de/fb12/en/index_html?set_language=en

people increase as said changes meet their interests, KONDA has chosen to collect requirements regarding the quality of cultural heritage data in a *bottom-up* process.

Although KONDA is a project that is based in Germany, the results yielded during the project's duration may be of interest to an international audience. Thus, we decided to publish all of our findings in English.

During our work with the quality problems gathered from different sources, we agreed on listing problems instead of requirements as in our opinion it might be more useful for our future work within the project context: Quality problems are something we can specifically observe in the data while requirements are a more abstract concept. Since our work is based on data (sub-)sets of two institutions , problems are likely to strike the eye, e.g. during the instantiation of the quality management process to be developed. Structuring our findings in problems rather than requirements simplifies this instantiation process since we do not have to work *ex negativo*, deriving which requirement is infringed from a problem at hand. The document can, therefore, be read as requirements specification since we defined a target state for problem description.

Moreover, by providing a catalogue of quality problems that occur in data, data models and data transformations we want to create tools for identifying specific problems and can advise on how a problem could be solved so that this deliverable could also be published (after slight alterations) for external usage.

To gather the requirements that serve as a basis for our further work, we took three different methodological approaches:

- We conducted **interviews** with different domain experts that perform acquisition, modelling, management and usage of various kinds of cultural heritage data, and asked them about their opinion and experiences with data quality in the cultural heritage sector. A questionnaire was prepared for each stakeholder group to guide the conversations. The interviews were designed openly and the questions were only for guidance. The overall aim of all questions was to detect and discuss quality problems. For each group, a list of well-known experts from the community was created, prioritized and written to individually. The prioritization takes place with the help of people from the community and the final selection was made by the project members. The experts were guaranteed complete anonymity. This means that the names are only known within the project and there should be no assignment of the quality problems to the persons. All interviews were recorded and transcribed internally by the project members. Both the audio files and the transcriptions will not be published for data protection reasons. The most important step was to analyze the transcriptions from which the quality problems in this document were extracted.
- We held two **workshops** within the broader cultural heritage community about quality requirements where people from different GLAM institutions have been invited to. The selection was made similar to the interviews by prioritizing the community and final selection by the project members. The workshops were organized as face-to-face meetings and carried out in various rounds of discussion on six different topics in a world café format. Questions were prepared for each topic. The questions were designed openly to generate free discussions by the participants. The aim was to identify and discuss topic-related quality problems and solutions. The results were protocolled and evaluated by the project members. For reasons of data protection, we did not capture any connections between quality problems and people.

- We **analyzed data** from the Centre for Collection Development of Göttingen University[4] and the Deutsches Dokumentationszentrum für Kunstgeschichte - Bildarchiv Foto Marburg[5] and extracted quality problems from them. We got access to nearly 30,000 data records of the Centre for Collection and over 800,000 data records from Foto Marburg with additional 360,000 records containing supplementary information. Hereby we especially focussed on amounts of individual values in the data.

Accompanying these approaches we also surveyed the current research regarding quality problems in data models, data transformations and data itself.

1.3. Underlying Data and Data Models

In each quality problem profile, concrete examples are given through data excerpts present in the LIDO[6] or MIDAS[7] format.

LIDO (Lightweight Information Describing Objects), a specific CIDOC-CRM[8] application, is an event-based XML format for describing museum or collection objects. Memory institutions use LIDO for "exposing, sharing and connecting data on the web"[9]. It can be applied to all kinds of disciplines in cultural heritage, e.g. art, natural history, technology, etc.

MIDAS (Marburger Inventarisations-, Dokumentations- und Administrationssystem) is an object-based XML format for the description of art-historical objects and related entities, such as artists and ateliers, developed by Bildarchiv Foto Marburg[10].

The MIDAS-Dataset provided by Bildarchiv Foto Marburg represents various cultural heritage objects, like architecture, paintings or sculpture. Created within the context of the archives work, the data set also contains information on (mainly) documentation photography depicting a given work of art.

For online publication on the Website "Bildindex" serving as a database for images and information on cultural heritage objects, MIDAS-Data is being transformed into LIDO-XML. The Bildindex also provides data from other institutions. Therefore the LIDO-Dataset provided by Foto Marburg

[4] Centre for Collection Development of Göttingen University. https://www.uni-goettingen.de/en/440706.html (Accessed 2020-08-05)

[5] Deutsches Dokumentationszentrum für Kunstgeschichte - Bildarchiv Foto Marburg. https://www.uni-marburg.de/de/fotomarburg (Accessed 2020-08-05)

[6] Erin Coburn, Richard Light, Gordon McKenna, Regine Stein, Axel Vitzthum. „LIDO (Lightweight Information Describing Objects)". http://network.icom.museum/cidoc/working-groups/lido/. (Accessed 2020-08-04)

[7] Jens Bove, Lutz Heusinger, Angela Kailus. Marburger Informations-, Dokumentations- und Administrations-System (MIDAS): Handbuch. 2. Aufl. Literatur und Archiv 4. München: Saur, 1992. https://doi.org/10.11588/artdok.00003770. (Accessed 2020-08-04)

[8] Martin Doerr, George Bruseker, Chryssoula Bekiari, Christian Emil Ore, Stephen Stead, Thanasis Velios. „CIDOC Conceptual Reference Model (CRM)". http://www.cidoc-crm.org/. (Accessed 2020-08-04)

[9] Erin Coburn, Richard Light, Gordon McKenna, Regine Stein, Axel Vitzthum. „LIDO (Lightweight Information Describing Objects)". http://network.icom.museum/cidoc/working-groups/lido/. (Accessed 2020-08-04)

[10] Germany's documentation center for art history, Deutsches Dokumentationszentrum für Kunstgeschichte – Bildarchiv Foto Marburg. https://www.uni-marburg.de/de/fotomarburg (Accessed 2020-08-04)

contains the transformed MIDAS-Data as well as LIDO-Data of other Archives, Museums and Libraries.[11]

Furthermore, we included a dataset provided by the Centre for Collection Development of Göttingen University. This dataset represents cultural heritage objects collected within the scope of various academic fields like Ethnology, Archaeology, History, Biology, Physics or Mathematics. Therefore a multitude of material objects from prints to medical instruments is described in the data.

1.4. Structure of the Profiles for Quality Problems

The catalogue describes 55 data quality problems, 13 data model quality problems and 3 data transformation quality problems. Other problems, e.g. discussed in the interviews or during the community workshops, are out-of-scope.

To get a standardised overview of each quality problem regarding data, data models and transformations, we decided to develop a profile that encompasses all dimensions that are of interest and are necessary for our further work. These include:

- A brief **description** which summarises the problem
- The **mainly affected quality dimension**. These quality dimensions have been defined within the project for each of the areas to be approached (data, data model, data transformations). Some dimensions are divided into subcategories that are marked with brackets in the profile template. The dimensions and their definitions are another result of the project and will be published separately.
- **Other affected quality dimensions**
- A description of how this problem **impacts data quality**
- One or more **examples** that illustrate the problem. These examples were taken from real data or constructed if no real data has been at hand.
- A list of possible **causes** that have led to this quality problem
- In which **stage of the research data cycle** this problem occurs or takes root. The stages are oriented towards the research data life cycle of the RfII[12].
- How this problem can be **identified** in a specific data record or database. This can encompass (semi-)automatic approaches as well as workflows.
- How a data record or database should look like when this problem has been solved (**target state**). **This part of the profile includes the actual requirements.**
- Which **preventive** measure(s) can be undertaken to avoid the described problem in the first place. These measures are approaches that are based on the interviews, the community workshops and the current state of research. This creates the basis for further work within the project. Therefore, existing tools and workflows will be analyzed and tools and workflows will be developed.
- Which **retrospective measure(s)** can be undertaken if the problem can already be located in a data record or the database. These measures are based on the interviews, the community workshops and the research of the project on the current state of research. This creates the basis for further work within the project. Therefore, existing tools and workflows will be analyzed and tools and workflows will be developed.

[11] Bildindex der Kunst und Architektur: https://www.bildindex.de/cms/homepage/ueber-uns/partnerinstitutionen/archive-bibliotheken-verlage/ (Accessed 2020-08-05)

[12] German Council for Scientific Information Infrastructures (RfII): The Data Quality Challenge. Recommendations for Sustainable Research in the Digital Turn, Göttingen 2020, 120 p. http://www.rfii.de/?p=4203. (Accessed 2020-08-05)

1.5. Profile Template for Quality Problems

<ID> <Problem_Name>

Description: <What exactly is the problem?>
Mainly affected quality dimension: <choose one of the categories below>
Other affected quality dimensions:
<for data problems: accuracy, consistency (consistency, uniqueness), usability (understandability, relevancy, trustworthiness, timeliness, accessibility (interoperability, quotability)), reusability, completeness (precision)>
<for model problems: correctness, completeness (precision), understandability, simplicity, changeability, implementability, uniqueness, modularity>
<for transformation problems: understandability, modifiability, reusability, modularity, completeness/correctness, consistency, efficiency, conciseness, reliability, interoperability>
Impact on data quality: <Why and how does the problem decrease data quality?>
Examples: <Data example, possibly MIDAS or LIDO>
Causes: <Why and under which circumstances does this problem occur? (e.g. model structure, system, practice, human error)>
Root in the data life cycle:
<plan, collect, assure, describe, submit, preserve, discover, integrate, analyze, publish>
Identification: <How can this problem be located in datasets? (e.g. pattern, metrics)>
Target state: <What should be achieved when fixing this problem?>
Preventive improvement: <How can this problem be avoided prematurely for new data records?>
Retrospective improvement: <How can this problem be handled? How can the quality of affected records be improved?>

1.6. The Version of the catalogue and future work

This is the first version of this catalogue (1.0). Since research within the KONDA project is progressing we like to keep the catalogue open for changes and enrichments of its content. The project's iterative structure allows us to collect further requirements. Also, we like to be able to include feedback from an international community. For feedback or comments on this document please contact: konda@uni-marburg.de

2. Quality Problems Concerning Data Models

This section encompasses all quality problems that are connected to the design of the data model. According to our definition, we developed at the beginning of the KONDA project, a data model is a model for the structured organization of the data of an application area. It is a formalization of the set of entities to be described, their properties and relationships to other entities represented in the data. A distinction is usually made between conceptual, logical and physical data models.[13] [14] [15]

MODEL01 Incomplete Data Model

Description: Some information cannot be stated in the data model, generally because of missing fields. This includes all problems that occur when the data model is unable to capture the described objects in the required precision.

Mainly affected quality dimension: Completeness

Other affected quality dimensions: Correctness, precision

Impact on data quality: Although more knowledge about an object is available not all of it can be placed in the data since the respective fields are missing. This may even lead to the data being useless to a user.

Examples:

- It is not possible to model all necessary entities, e.g. both cultural heritage and scientific objects
- It is not possible to provide information obtained from oral statements (e.g. calls, talks, conversations, ...) as source
- A specific field is not defined as repeatable, but there are cases where multiple values shall be assigned in a data record.

Causes:

- The data model has been developed without (proper) requirements engineering
- Requirements have changed over time
- No participation of domain experts during the development of the data model

Root in the data life cycle: Plan

Identification:

- Extensive use of free text fields
- Misuse of elements/statements (cf. DATA02.4)

Target state: All required information related to an entity can be provided in specific data fields in a structured way (i.e. without using free text).

Preventive improvement:

- Thorough requirements engineering
- Involve domain experts during data modelling to check if all necessary information can be depicted
- Iterative assessment of data model: Does it still fit the institution's requirements?

Retrospective improvement: The data model could be enhanced and missing data be added. This procedure would be tied to a lot of work and resources though since everything has to be added and checked manually.

---

[13] Paulo Merson: Data Model as an Architectural View. 2009. Software Engineering Institute, Carnegie Mellon University, 10.1184/R1/6572864.v1 (Accessed 2020-08-04)

[14] Elmasri, Navathe: Fundamentals of Database Systems, .3 edition, Verlag: Pearson Studium, ISBN: 978-0-136-08620-8, 2011

[15] Saake, G., Sattler, K.-U., Heuer, A.: Datenbanken - Konzepte und Sprachen, 4. Edition. MITP. ISBN: 978-3-8266-9057-0, 2010

MODEL01.1 No Value-Exclusions (“not B”)

Description: For some data models it is not possible to mark explicit exclusions of values.
Mainly affected quality dimension: Accuracy
Other affected quality dimensions: Completeness
Impact on data quality: In some cases, we can restrict a set of possible values by excluding certain elements. When a data model does not allow for expressing the exclusion of certain values, another form of modelling “possibly X, Y, Z but not A” has to be provided. If this is not the case, knowledge in the form of “A is not the case” is lost, leading to a loss of information.
Examples: “I do not know, who created this, but it was certainly not artist X!”
Causes:

- The data model does not provide the possibility to express value-exclusions

Root in the data life cycle: Plan
Identification:

- Since this information is possibly provided in a free text field: Querying them for keywords (e.g. “not”)
- If information is omitted these cases cannot be identified automatically but only with human intellect/expert knowledge of the cataloguers

Target state: It should be possible to express knowledge of the form “possibly X, Y, Z but not A” or “not A” in the data when there is evidence that “A” is not the case.
Preventive improvement: Consider modelling an exclusion statement or element when developing the data model.
Retrospective improvement:

- Add exclusion statement or element the data model
- Identify data sets coming into question. In some cases knowledge in the form of “possibly X, Y, Z but not A” may be retained in a free text field, making it possible to extract the data with textual analysis. In other cases, the information may not have been recorded during the acquisition, so that the respective data sets have to be identified manually (time-consuming and costly).

MODEL02 Unspecific Field Concepts

Description: The description of single fields of a data model is so unspecific that it leaves much room for interpretation. Thus multiple information can fit. Sometimes this is wanted, like in an optional comment field, but for other data fields, an unspecific definition impedes comparability between data sets.

Mainly affected quality dimension: Uniqueness

Other affected quality dimensions: Consistency, understandability

Impact on data quality: An unclear specification of desired data generates very heterogeneous data values regarding subject and precision (DATA01.6.2). This can result in MODEL11.

Examples:

- MIDAS “2730 (Stelle)” (location) vs. ”5235 (Position)” (position)
- MIDAS “5240 (Formtype)” (a type of shape) allows highly specific terms (with AAT mapping, e.g. “hall church”) as well as general concepts (e.g. round)
- MIDAS “5365 (Maßzahl)” (What is meant by “measurement”? height? length? weight?)
    - The type of measurement must be specified in “5364 (Maßart)”

Causes:

- Unclear concept of fields in the data model

Root in the data life cycle: Plan

Identification:

- Confusion and insecurities: Cataloguers are unsure about what to
- Heterogeneous data

Target state: All fields/elements/statements in the data model are clearly defined.

Preventive improvement:

- Develop a clear model concept with specific rules for data fields
- Proper training of data acquisitors

Retrospective improvement:

- Redefine fields / add new rules for acquisition?
- Train staff wtr. new field/element/statement definitions

MODEL03 Use of Free Text Fields Instead of LOD References

Description: Information is provided in free text fields (and therefore filled in manually). The concept of the field within the data model is not specific enough. Instead of providing a reference to a Linked Open Data object or authority file, it is not possible to exactly identify the given information with a LOD-Reference.[16] Therefore the description or term can not be identified by a machine.

Mainly affected quality dimension: Precision

Other affected quality dimensions: Completeness, correctness, simplicity, implementability

Impact on data quality: In case information about e.g. a person is provided manually instead of relying on LOD/authority files, the way information about the said person is provided is possibly heterogeneous when cataloguers use different phrases or styles of description (cf. DATA01.6.4). To avoid this, manual editorial corrections are necessary. Furthermore, changes in information about e.g. a person have to be added manually as well while using LOD/authority files could lead to an automatic update as soon as the new information is added in the authority file system. Also providing information in free text fields is an unstructured way of information distribution which makes reusing the data more difficult.

Examples:

```
<lido:objectWorkType>
        <lido:term>Kupferstich</lido:term>
</lido:objectWorkType>
```

instead of

```
<lido:objectWorkType>
        <skos:Concept rdf:about="http://vocab.getty.edu/aat/300041341">
                <skos:prefLabel xml:lang="en">copper engravings (visual works)</skos:prefLabel>
                <skos:prefLabel xml:lang="de">Kupferstich (Druckgrafik)</skos:prefLabel>
        </skos:Concept>
</lido:objectWorkType>
```

Causes:

- Using LOD/authority files is not possible due to the data model (e.g. no URIs[17] allowed)
- Using LOD/authority files is not implemented in the cataloguing software
- Lack of knowledge: LOD/authority files generally unknown; which vocabulary is suitable for which use case?

Root in the data life cycle: Plan

Identification:

- Check if a URI is provided (REGEX)[18]

Target state: All information in a record should be linked to LOD objects/authority files if possible.

Preventive improvement:

- Implement the possibility to provide URIs in the data model
- Require data acquisitors to provide a URI for certain fields that make sense (or make them explain why this cannot be done)
- Staff training

Retrospective improvement: For a set of given elements/statements an automatic process using REGEX could check if a URI is provided. This set has to be identified manually in the first place which could lead to some overhead. Linking information to the correct authority data requires domain-specific knowledge but can probably be done easily in most cases.

[16] LOD = Linked Open Data: https://www.w3.org/wiki/LinkedData. (Accessed 2020-08-24)

[17] URI = Uniform Resource Identifier: https://en.wikipedia.org/wiki/Uniform_Resource_Identifier. (Accessed 2020-08-24)

[18] REGEX = Regular Expression: https://en.wikipedia.org/wiki/Regular_expression. (Accessed 2020-08-24)

MODEL04 Structural Redundancy in Data Models

Description: There can be multiple model elements with the same meaning. A data model allows for encoding the same fact in multiple different ways.

Mainly affected quality dimension: Uniqueness, precision

Other affected quality dimensions: Simplicity, changeability, consistency, uniqueness, understandability

Impact on data quality: By encoding the same fact or phenomenon in different ways a data set becomes heterogeneous and less precise. An application using the data has to consider all possible options. If there are different elements or relations this can lead to overhead in the application and elements/relations being overlooked.

Examples:

- A schema allows for tei:name and tei:persName at the same time, both being used for encoding a person's name[19]
- Events are structured differently between to data sets
- MIDAS does not have an XML specification
- Non-compliance of the given and intended structure

Causes:

- Human error during data model creation: Ambiguous elements/relations got overlooked
- The data model describes the complex and/or diverse entities and has to be that flexible
- Specification of the structure of the data model is not specified

Root in the data life cycle: Plan

Identification:

- Elements/predicates/fields with similar meaning
- Different sub-structures of the same element

Target state: The data model should be so flexible that everything necessary can be encoded while not allowing for heterogeneous data or imprecisions. In most cases, the model designer has to come to a compromise between model flexibility and heterogeneity.

Preventive improvement:

- Test drive for data model: By letting several data acquisitors create data records for similar objects in a test stage the model creator can evaluate if the model is too flexible
- Recommendations and indexing guidelines for data acquisitors to achieve a certain degree of streamlining

Retrospective improvement:

- Make data more homogeneous with semi-automatic scripts (automatic scripts may be too prone to errors)
- Analyze data set to determine how much do the data sets differ and correct them manually (time-consuming, high effort)

[19] TEI: Text Encoding Initiative https://tei-c.org/. (Accessed 2020-08-24)

MODEL05 Changes of Data Model Structure

Description: In the lifetime of a data model its structure can change. Hereby single fields can be added, deleted or redefined as well as moved to other points in the structure. This has effects on existing data records, which do comply with older versions of the data model.

Mainly affected quality dimension: Consistency

Other affected quality dimensions: Understandability, Changeability

Impact on data quality: While the data model evolves an institution has to ensure that older data records still comply with the new data model; Otherwise older data records are not usable anymore or may become wrong in case of element redefinitions. A new data model can also decrease the reusability of the data when third-party software still uses an older version of the data model. Therefore it is necessary to document the used data model version in the data (DATA07.2).

Examples:

- MIDAS has grown evolutionarily
- LIDO v1.1 provides additional separate rights information for object description, which LIDO v1.0 does not

Causes:

- Changing requirements
- Problems in data modelling that make changes necessary
- New technical possibilities

Root in the data life cycle: Plan

Identification: Structurally heterogeneous data

Target state: Data should be structurally homogeneous and comply with the newest version of the data model.

Preventive improvement:

- Requirements engineering before developing the data model
- Make changes in the data model that are backwards compatible (if possible)
- Develop a workflow for change management/data update management
    - Scripts
    - Workflows for manual changes

Retrospective improvement:

- Added elements: Retrieve data that lacks the new element and manually add missing information (time-consuming for large data sets)
- Deleted elements: Delete resp. elements with a script (relatively low effort)
- Redefined elements: Check if data with resp. element complies to the new definition (time-consuming since it requires human intellect)
- Elements moved to other places: Move resp. elements in a data set with a script (relatively low effort)

The scripts developed for the first retrospective improvement can be modified and re-used for future change management.

MODEL06 Incomprehensible (Complex) Model

Description: If the data model is used for describing many different phenomena in great detail the model has to be very large and also complex. The more complex and multifaceted a data model is, the harder it is to understand it.

Mainly affected quality dimension: Understandability

Other affected quality dimensions: Simplicity

Impact on data quality:

- If the person editing a data record does not fully understand the data model, the data s/he creates may be of bad quality by using the model incorrectly (cf. MODEL09, DATA02.4). Thus, information is expressed in inappropriate constructs or within the wrong fields.
- Additionally, high complexity of a data model may impact the time needed to create, edit or even read a data record.
- Complex and multifaceted data models complicate data exchange and their mapping into other models.

Examples:

- MIDAS does have over 2700 field types, from which only ca. 900 are used in the database of the DDK. Some fields have the same description, but use different numbers (serving as field names in MIDAS), depending on where in the document structure it appears. The model would be easier to comprehend if the same field names were used where the same type of information is expected.
    - Example: For “Ort”, there are 6 different fields, which should be used in different parts of the model (2464, 2664, 2864, 5108, 7070, o2664).

Causes:

- Model is intended to be used for describing many different kinds of entities or phenomena in great detail
- Model grows over time

Root in the data life cycle: Plan

Identification: (see Genero et al. 2005, Arendt Diss 2014 on class model metrics)

- Number of entities
- Number of relations (per entity)
- Number of attributes (per entity)
- Hierarchy depth

Target state: The data model should not be more complex than necessary to describe the entities of interest. Even users with little experience in data modelling and knowledge about the domain should be able to understand and use the model after a quick introduction and training phase.

Preventive improvement:

- Precisely specify the use cases before designing a data model
    - The model should only include constructs needed for these use cases
    - Include as few entities, attributes and relationships as possible
    - Avoid unnecessary generality
    - Avoid unnecessary precision
- Avoid any redundancy in the data model
- Reuse familiar concepts when designing the data model/rely on established design patterns where possible

Retrospective improvement:

- Data model refactoring: Behaviour preserving model transformations (Sunye et al. 2001)
    - See **Preventive improvements**

MODEL07 Model Makes Inaccurate Statements About the Domain

Description: The model does not comply semantically to the domain. The model structure (order of data fields) implies dependencies and relations, which do not comply. E.g. in XML the model hierarchy is badly ordered, when more general or independent fields are represented as more specific or dependent. That structure, therefore, implies the wrong relationships between fields.

Mainly affected quality dimension: Correctness

Other affected quality dimensions: Uniqueness, understandability

Impact on data quality: Values are not well placed which complicates the data presentation and the overview when using the data.

Examples:

- Wrong direction of relation: Holder of the object is Saarlandmuseum, but the object can't be the holder of Saarlandmuseum.
- In XML bad hierarchy: No nesting of dependent values

```
<object>
        <title>Gemüsestilleben</title>
        <name>Max Slevogt</name>
        <person>
                <GND-ID>http://d-nb.info/gnd/118614940</GND-ID>
        </person>
</object>
```

instead of:

```
<object>
        <title>Gemüsestilleben</title>
        <person>
                <name>Max Slevogt</name>
                <GND-ID>http://d-nb.info/gnd/118614940</GND-ID>
        </person>
</object>
```

- In XML bad hierarchy: Nesting of independent values

```
<object>
        <title>Gemüsestilleben</title>
        <person>
                <name>Max Slevogt</name>
                <GND-ID>http://d-nb.info/gnd/118614940</GND-ID>
                <term>painting</term>
        </person>
</object>
```

instead of:

```
<object>
        <title>Gemüsestilleben</title>
        <person>
                <name>Max Slevogt</name>
                <GND-ID>http://d-nb.info/gnd/118614940</GND-ID>
        </person>
        <term>painting</term>
</object>
```

Causes:

- Bad design
- No consultation of experts during and after the modelling process

Root in the data life cycle: Plan

Identification: Manual

Target state: All dependencies and relations of values are represented in the model structure

Preventive improvement:

- Good design planning
- Consultation of experts before and during modelling phase

Retrospective improvement: Restructure the data model manually

MODEL08 Too Few Mandatory Fields

Description: There are too few mandatory fields in the data model to get a real impression about the object when all optional fields are left empty.

Mainly affected quality dimension: Completeness

Other affected quality dimensions: Understandability

Impact on data quality: Even though a data record is available for an object, the record is not usable for anything other than stating that this object exists.

Examples:

- In LIDO v1.0, only the object type, the object's title, the record ID, record type and record source are mandatory

Causes: Design of the data model

Root in the data life cycle: Plan

Identification:

- Assessing the data model for flags or cardinality that imply mandatory fields

Target state: There are enough mandatory fields to gain a thorough impression of an object described in the record. Useful fields may vary for different institutions or object types.

Preventive improvement:

- Thorough requirements engineering: What is the minimal information content a record should encompass to make it useful for users?
    - Workshops, UX personas, ...

Retrospective improvement: The data model could be changed when the necessity arises, but since these changes are most likely not backwards compatible, all existing data records would need to be updated (worst case scenario: Manually).

MODEL09 Missing Documentation of Data Model

Description: The data model is not properly documented.
Mainly affected quality dimension: Understandability
Other affected quality dimensions: Correctness, completeness
Impact on data quality: A data model not properly documented impedes its use on all levels, since element definitions may be unclear (which leads to problems in the implementation into acquisition software), decisions for a certain design may be obscure (making it less extensible) and due to the possibly reduced understandability it is less reusable. Furthermore, the data model could be misused for other purposes or scopes. For example, one class of the data model could be used for another entity.
Examples:

- An XSD without xs:documentation
- Using a LIDO profile for a different scope than it was designed or intended for
- Entities are found under the wrong class

Causes:

- Lack of resources or time during the data model design
- Model designer regards the model as self-explanatory
- Designer unfamiliar with best practices

Root in the data life cycle: Plan, collect
Identification: No documentation (either as separate file/website or documentation elements/strings) is available in the data model.
Target state: A data model should provide documentation about itself.
Preventive improvement:

- Taking time for writing docs into account when planning resources for the data model
- Peer-review model creation and changes and require docs in this process
- Train data model designer wrt. best practices

Retrospective improvement:

- Add missing docs. This is time-consuming since it has to be done manually.

MODEL10 Missing Unique Identifiers

Description: A data record lacks a unique identifier by which it can be differentiated from other records.

Mainly affected quality dimension: Uniqueness

Other affected quality dimensions: Precision, consistency, uniqueness, understandability, implementability

Impact on data quality:

- Data records are not identifiable.
- DATA 05.3

Examples:

- <lido:recordID/>
- Records from different sources contain the same recordID describing different entities.

Causes:

- The data model does not provide field/element/statement
- IDs are not generated automatically and forgotten during data acquisition

Root in the data life cycle: Plan

Identification: Data records lack unique values by which they could be differentiated from other records.

Target state: All data records should be identified based on one specific value which serves as an identifier.

Preventive improvement:

- Implement field/element statement for ID into the data model
- Automatically generate and assign an ID to the new data record
- Ensure uniqueness of the ID

Retrospective improvement:

- Remove equal values between data records which are used for references.
- Enable ID allocation in the data model
- Automatically assign IDs to all existing data records
- Update existing references so that they address IDs instead of other values. Possible data records can be queried automatically while some human intellect may be necessary in case the reference is not unique.

MODEL11 Multiple Information Encoded Together in a Single Field

Description: The data model requires multiple different kinds of information about the described entity to be encoded together in a single field. They are separated e.g. by commas or space.
Mainly affected quality dimension: Uniqueness
Other affected quality dimensions: Understandability
Impact on data quality: Values are less machine-readable
Examples:

- MIDAS “5365 (Maßzahl)” (a combination of height, width, length or weight with its unit)
- MIDAS “5385 (Art und Zahl)” (type and number)

Causes:

- Simple models with a small number of fields
- The data model does not provide multiple fields for that information

Root in the data life cycle: Plan, collect
Identification:

- The data model documentation lists multiple information aspects that should be encoded in the field
- Regex: Detect the use of delimiters or long entries

Target state: Different kinds of information should be represented in multiple different fields.
Preventive improvement:

- Separate value-pairs in multiple fields

Retrospective improvement:

- Modify the data model by adding fields
- String Analysis of bad input and divide entry into multiple fields

MODEL12 Inappropriate Primitive Data Types

Description: The data model does not specify appropriate primitive data types for corresponding fields. The values that can be entered in a specific field are not limited adequately. Often many fields are specified to be of type string even though they represent numerical or temporal information.

Mainly affected quality dimension: Understandability

Other affected quality dimensions: Precision, correctness

Impact on data quality: Data quality is decreased if the type of the expected information is not specified precisely as this enables unwanted (i.e. unexpected and invalid) values to be entered in this field. This prevents the direct comparison of values given in this field across records.

Examples: In MIDAS every primitive field is of type string

Causes:

- Data model designers want to allow the flexible, implicit encoding of additional information in the same field instead of modelling this additional information explicitly

Root in the data life cycle: Plan

Identification:

Target state: By specifying an appropriate primitive data type the accepted values for a field should be limited as far as possible. Each field should represent exactly one information aspect.

Preventive improvement:

- Conduct a requirements analysis for the data model
- Model required information explicitly

Retrospective improvement:

- Improve data model and transfer existing data to the new model via pattern-based transformation

3. Quality Problems Concerning Data

This section encompasses all quality problems that are connected to the data itself, i.e. the respective data record instances. As a data record, we define a certain amount (1-n) of data fields that are connected through their content describing one entity.

DATA01 Data Volume

Problems in the context of the quantity of data.

DATA01.1 Missing Data

DATA01.1.1 Missing Data - Empty Fields

Description: Fields in the data record are empty, thus the expected information is not given. Application profiles may be used for specifying mandatory and optional fields.
Mainly affected quality dimension: Accuracy
Other affected quality dimensions: -
Impact on data quality: Empty fields decrease data quality since they imply that information about the described entity is missing from the record but do not further specify why it is missing (e.g. information is not known according to the current state of research vs. not enough time for research when creating the record).
Examples:

- <lido:subjectConcept>
  <lido:term/>
  </lido:subjectConcept>

Causes:

- Human error and insecurities of Cataloguers
- Part of the information cannot be determined with up-to-date methods and by evaluating all available sources
- Part of the information is unknown to the person editing the data record – the resources may not be sufficient for overcoming gaps in knowledge through research
- Resources are not sufficient for transferring all known information to the data record
- Information is ignored as it is being misinterpreted, misjudged, differs from expectations or the person does not know how to handle it
- The person editing the data record does not understand the field's specification

Root in the data life cycle: Collect
Identification:

- Empty strings
- Missing statements

Target state: The data should describe the entity of interest as completely as possible according to the current state of research. Mandatory fields should not be empty. In the case of empty fields, the reason for it to be empty should be specified (DATA06.7).
Preventive improvement:

- Do not allow to finally save a new record before all mandatory fields are filled
  - Necessary to create a well-considered, balanced system of mandatory fields
- Encourage cataloguers to make uncertain statements rather than none at all by allowing uncertain statements to be marked and qualified appropriately in the record
- Design data model
  - Such that the cause for missing data can be specified as well as the state of acquisition - take into account the impacts on the organization's reputation
  - Such that research desiderata can be made explicit

        - Allows to call for crowdsourcing of external expertise
- Show percentage of empty fields to the person editing the record
- Specify the purpose of each field in an understandable manner
- Template for new acquisition based on a minimal desired data record

Retrospective improvement:

- Update records as soon as new research results become available
    - Development of workflow to gather new research findings into data
- Review records with many empty or any empty mandatory fields

DATA01.1.2 Missing Data - Incomplete Fields

Description: Fields are only partially filled. Not all desired information is stated in the model. Fields can be incomplete, due to lack of knowledge (only partial information) or formal mistakes (missing units). Hence there is not always a possibility to complete those fields.
Mainly affected quality dimension: Completeness
Other affected quality dimensions: (accuracy, uniqueness, usability)
Impact on data quality: Information about the described entity is missing from the record.
Examples:

- No exact date for points in time, only a year
- Number without units (DATA04.2)

Causes:

- Lack of research knowledge
- Implicit encoding of uncertainty (e.g. imprecision)
- Human error: Work has been interrupted and not been completed

Root in the data life cycle: Collect
Identification: Fields do not contain all the desired information.
Target state: All fields meet their requirements on information density.
Preventive improvement:

- During acquisition: REGEX for input
- Clear acquisition workflow for entry of data fields

Retrospective improvement:

- Add manually after automatic identification

DATA01.1.3 Missing Records

Description: Records that should be in the database are missing due to various reasons.
Mainly affected quality dimension: Completeness
Other affected quality dimensions: Usability
Impact on data quality:

- When a record is missing in the database, the data set is not complete (though this might be a permanent state due to acquisition practices). Thus queries over the full database might lead to erroneous results and it is hardly possible to defer statements like “in the collection X, Y is the case”. Furthermore missing data records might cause a loss of trustworthiness and reputation of an institution on the user side, e.g. when a user knows a specific drawing is held in the institution but the data record is missing.
- Information about the related data record is missing. References cannot be resolved and are unclear, thus impeding further use of the still existing record. (DATA05.2)

Examples: A painting is displayed in a museum but it is missing in the museum’s database.
Causes:

- Practice: Cataloguing can only start after the acquisition of an object. The needed time for data creation depends on the resources.
- Technical: Data loss, database index not updated/not working, wrong references, ...

Root in the data life cycle: Collect, publish
Identification: If there are references to an object that does not have a data record, this might be discovered by an unsuccessful resolving query. Otherwise, the lack of a data record is either known (or discovered by chance) or not.
Target state: Each object should have one data record.
Preventive improvement:

- Create lists of an institution’s (known) objects in the inventory with preassigned IDs that are used during cataloguing (thus creating a mapping)

Retrospective improvement:

- A mapping between a list of data records and an institution’s (known) objects (possibly time-consuming if there are no digital inventory lists or they are not suitable for comparison)

DATA01.1.4 Missing Source

Description: In a data field an interpretative (i.e. non-measurable) statement is made without providing a source (e.g. a signature, historical document, (scientific) paper, Wikipedia, the expertise of the cataloguer) for it.

Mainly affected quality dimension: Trustworthiness

Other affected quality dimensions: Accuracy

Impact on data quality: For the traceability and trustworthiness it is necessary to specify the source of acquired and registered information.

Examples:

- "'The Man with the Golden Helmet' is no longer attributed to Rembrandt since 1985" without providing a source for this information
- "Some scholars state that XYZ" without providing a source for this information

Causes:

- Sources are provided in a non-structured way, e.g. in a free text field
- Acquisition software does not provide an input field for sources
- The data model does not allow for providing sources
- A statement (though being interpretative) is widely accepted as *communis opinio* and thus no source is provided
- Providing sources is out of scope in an acquisition project

Root in the data life cycle: Collect

Identification:

- By validation, if providing sources is mandatory in the data model (low effort)
- Searching free text fields (high effort)

Target state:

- It should be possible to provide sources for certain data fields or relations between entities. Preferably all statements should be verified by a source.

Preventive improvement:

- Data field for sources in the acquisition software
- Use authority data for referring to sources. This also makes data acquisition much easier.
- Visual cue (like a warning triangle, ...) in software if sources are optional

Retrospective improvement: Cannot be done with a reasonable effort since data aggregators probably do not know where to search for the information already provided. Adding sources in a later stage is very time-consuming.

DATA01.1.5 Person Responsible for Uncertain Statements Missing

Description: If uncertain statements are made that are not based on proper sources and thus do not necessarily represent the current state of research but rather a single person's subjective interpretation, the person (e.g. cataloguer) who is responsible for making a said statement should be denoted in the data. Since this person then serves as the source for the statement, this is a special case of DATA01.1.4.

Mainly affected quality dimension: Completeness

Other affected quality dimensions: Precision, trustworthiness

Impact on data quality: In case statements are uncertain, e.g. because the knowledge stated is inferred without documented evidence, it should be traceable who has decided to make this statement. Otherwise, the data might suggest uncertainty as to general "knowledge". Also providing the person responsible leads to better transparency and therefore increases the trustworthiness of the data.

Examples:

- Assume it is highly probable that an object has been created at a certain location but there is no definite evidence (e.g. evidence by documents). Nevertheless, the place of creation is stated in the data, but no clue has inferred said place.

Causes:

- The data model does not allow for providing authors (e.g. cataloguer)
- Author statement has been forgotten during the acquisition
- Author statements are out of the scope of the acquisition project

Root in the data life cycle: Plan, collect

Identification:

- Elements, attributes or statements that describe uncertainty lack an additional element, attribute or statement for the statement's author

Target state: All statements that are marked as uncertain should provide an author or source for the uncertain information.

Preventive improvement:

- Consider authors of uncertain statements in data model development
- Train staff in this respect
- Implement cataloguing software such that it automatically inserts the current cataloguer as to the author (i.e. source) if no other source is stated

Retrospective improvement: Respective statements can easily be identified by a database query. However, it might be impossible to retrospectively recollect the statement's author. In this case, the author could be "unknown", "anonymous" or the like (which can be added automatically). This term should be controlled by a vocabulary.

DATA01.1.6 Missing Metadata

Description: Metadata about data records (e.g. which institution has created the data, licensing, identifier, ...) are not stated in the data. More precisely, the following types of metadata may be of interest but still be missing from the data: Time required for creating the record, level of expertise of the person editing the record, change history, provenience, the context of creation: project, focus, goal/purpose, limits, resources, respected quality dimensions, research question, obligatory fields ...

Mainly affected quality dimension: Completeness

Other affected quality dimensions: Usability, trustworthiness

Impact on data quality: Data records that lack meta-information e.g. about the data creator or license are not trustworthy or re-usable since a user cannot comprehend the data origin and possible usage. This may even lead to the data record to be completely useless. Missing metadata also makes it harder to reconstruct how and why data quality problems have emerged. This complicates quality improvement of old data.

Examples:

- <lido:recordRights> completely missing
- <tei:revisionDesc/> (no revision history stated)

Causes:

- Human error: Inattention
- Metadata elements not mandatory or missing at all
- The software does not warn the user when mandatory elements are missing
- Lack of time
- Legal issues, e.g. log files of acquisition software must not be published
- The institution does not want to disclose internal procedures and habits to the public

Root in the data life cycle: Plan, collect, describe

Identification:

- Check if (obligatory) metadata elements or statements are missing or left empty

Target state: All obligatory metadata elements or statements should be existent and filled in with correct information. Also, any metadata that could in any way help in understanding, assessing or using the data and that can lawfully be published should be included in the data.

Preventive improvement:

- Train staff
- Make metadata mandatory in the data model
- Highlight mandatory fields via a validation
- Restrict saving of documents to records where all obligatory metadata has been provided
- Metadata that can be tracked by tools should be provided technically instead of manually
- Regularly check if there are invalid data records

Retrospective improvement: Query database for records in which obligatory metadata elements or statements are missing and add the missing information. Depending on the kind of information missing this can be done effectively in a (semi-)automatic way or has to be done manually.

DATA01.1.7 Rights Statement and/or License Missing

Description: It is unclear which rights are tied to single data records if it is not stated directly in the data. Generally, for different data records, distinct rights are mandatory.

Mainly affected quality dimension: Usability

Other affected quality dimensions: -

Impact on data quality: Data records without a rights statement and/or license cannot be used for further publication, research etc. Thus the record is less valuable for other researchers, journalists, ... or even unusable.

Examples:

- <lido:recordRights>...</lido:recordRights> is missing

Causes:

- No respective element/statement for providing rights and licenses in the data model
- Human error: Providing rights is forgotten during data acquisition
- The legal situation is unclear due to the current status

Root in the data life cycle: Plan, Collect

Identification:

- Check for empty elements/malformed statements if providing rights and licenses is possible
- No elements/statements available

Target state: Every data record should hold information about its legal status and conditions of reuse.

Preventive improvement:

- Provide the possibility of creating information about rights and licenses connected to the record
- Make the resp. elements/statements mandatory
- Design data model such that unclear legal situations can be marked explicitly

Retrospective improvement:

If resp. elements/statements are available in the data model:

- Check for empty elements/malformed statements if providing rights and licenses is possible. insert the apt legal information (possibly time-consuming if the information has to be checked and added manually)
- Make the resp. elements/statements mandatory and check availability with validation

Otherwise:

- Update data model and provide the possibility of creating information about rights and licenses connected with the record
- Make the resp. elements/statements mandatory and check availability with validation
- Insert the apt legal information (possibly time-consuming if the information has to be checked and added manually)

DATA01.1.8 No Rating of A Source in Data

Description: Some sources for information are more trustworthy or convincing in their reasoning than others, but these gradations are not directly expressed in the data.

Mainly affected quality dimension: Precision

Other affected quality dimensions: Consistency, understandability, trustworthiness, completeness

Impact on data quality: If all sources are treated equally although some are more trustworthy/relevant/... than others, implicit knowledge of cataloguing persons is not made explicit. Furthermore, users cannot be sure which ones are "the right ones", i.e. the reliable sources that have been used during the data creation. This decreases the understandability and trustworthiness of the data.

Examples:

- A statement published in a widely accepted research journal is more trustworthy than a statement taken from Wikipedia
- The reasoning for a painting's dating introduced in paper A is more convincing than the one in paper B

Causes:

- The data model does not provide a rating of a source
- The staff does not want to take up a position due to lack of knowledge/undecidable cases/political reasons/...

Root in the data life cycle: Collect

Identification: Sources are provided but without an attribute/relation that weighs them.

Target state: In cases where data acquisitors can only provide a source for information that has a low degree of trustworthiness this should be emphasized because otherwise, the data suggests a level of trustworthiness/certainty, ... that does not meet reality. In case different opinions about a fact exist these should also be outlined in the data but the more trusted one(s) should be marked to ensure the traceability of the information.

Preventive improvement:

- Enable source rating in the data model
- Establish institutional workflows that address this problem
    - Individually rate every source vs. specify rating per type of source

Retrospective improvement: Sources without a rating could be identified easily in a database query. Rating them probably is time-consuming since all sources used during cataloguing have to be re-read and evaluated.

DATA01.5 Unmarked Multilingualism

Description: In a data set several languages are used for the same field without marking the respective language.

Mainly affected quality dimension: Precision

Other affected quality dimensions: Accuracy

Impact on data quality: While it is technically correct and desirable to provide descriptions and labels in different languages, not marking which language is used leads to heterogeneous data. Also, applications using the data depend on this information for correct localization.

Examples:

- <lido:measurementType>Breite</lido:measurementType> and <lido:measurementType>width</lido:measurementType> in another record of the data set without an @xml:lang to specify different languages

Causes:

- Default language unclear
- Human error: Attribute or predicate forgotten
- Data model cannot express multilingualism

Root in the data life cycle: Pan, collect

Identification:

- Probably by chance in most cases, especially when data is indexed in several languages.
- A (semi-)automatic approach may prove difficult since there are several options to express something so a search e.g. German words might not yield any results because the wrong wording has been chosen in the query.

Target state: A default language for the data set should be defined. Labels and descriptions in other languages have to be marked as such, e.g. by an attribute or an additional statement.

Preventive improvement:

- Proper documentation of encoding multilingualism in the acquisition guidelines
- Software support for cataloguing: Dropdown or the like to indicate a label is not composed in the default language

Retrospective improvement:

- An option would be to parse all labels and descriptions and compare them to a dictionary of a specific language. This might be prone to error, though, since domain-specific vocabulary might not be part of the dictionary.
- Furthermore, some terms, e.g. technical terms or original titles for the object are coined in a certain language (e.g. *impasto, chiaroscuro, ...*) and have to be tolerated.

DATA01.6 Heterogeneous Data

DATA01.6.1 Heterogeneous Structural Representations

Description: The same information is represented in structurally heterogeneous forms. This problem often occurs when integrating data from multiple sources.
Mainly affected quality dimension: Consistency
Other affected quality dimensions:
Impact on data quality:

- Data records are not as suitable for further use as they could be since applications processing them have to test for different structures, available data elements etc.

Examples:

- A measured value is represented as a result of an event in some cases and as an attribute of an object in other cases

```
<lido:objectMeasurementsSet>
        <lido:displayObjectMeasurements>Höhe x Breite: 130 x 110 cm</lido:displayObjectMeasurements>
        <lido:objectMeasurements>
            <lido:measurementsSet>
                <lido:measurementType>Höhe x Breite</lido:measurementType>
                <lido:measurementUnit>cm</lido:measurementUnit>
                <lido:measurementValue>130 x 110</lido:measurementValue>
            </lido:measurementsSet>
        </lido:objectMeasurements>
</lido:objectMeasurementsSet>
```

vs.

```
<lido:eventObjectMeasurements>
        <lido:displayObjectMeasurements>Höhe x Breite: 130 x 110 cm</lido:displayObjectMeasurements>
        <lido:objectMeasurements>
            <lido:measurementsSet>
                <lido:measurementType>Höhe x Breite</lido:measurementType>
                <lido:measurementUnit>cm</lido:measurementUnit>
                <lido:measurementValue>130 x 110</lido:measurementValue>
            </lido:measurementsSet>
        </lido:objectMeasurements>
</lido:eventObjectMeasurements>
```

Causes:

- MODEL04
- DATA09.1
- Dynamics in data model: Different versions of the model are used
- The scope of the acquisition context changes (CAUSE12)
    - Institution-specific practices
    - Varying research questions which influence on acquisition practices (project context)

Root in the data life cycle: Collect, Integrate
Identification:

- Identify MODEL04 and DATA09.1 representing the same meaning and check if more than one representation is used in the data via existence patterns and regex
- Compare number of occurrences of suspicious structural constructs between the data sets that are to be integrated

Target state: Information should be represented in a uniform way.
Preventive improvement:

- Design data model such that each type of information can be represented only in one structural form
- Manual data review before imports or integration
- Provide a guide for cataloguers with acquisition examples for different types of objects. These examples should reflect state of the art procedures like providing URIs to LOD / controlled vocabulary.

- Use the newest version of the data model

Retrospective improvement:

- Improve data model
- Decide for one unique representation and transform other representations (detected via patterns) into this one

DATA01.6.2 Heterogeneous Precision of Data

Description: In single data fields the precision of the value is very variable in the data record. Hence, the problem lies in the heterogeneity of the precision, not the imprecision itself (as described in DATA06.2).

Mainly affected quality dimension: Precision

Other affected quality dimensions: Usability

Impact on data quality: Applications and analyses based on the data may become difficult since the data is not comparable. Also, data may be too vague/disparate to be useful.

Examples:

```
<lido:event>
 <lido:eventDate>
        <lido:date>
        <lido:earliestDate lido:type="approximate">1553</lido:earliestDate>
        <lido:latestDate lido:type="approximate">1633</lido:latestDate>
        </lido:date>
 </lido:eventDate>
</lido:event>
```

vs.

```
<lido:event>
 <lido:eventDate>
        <lido:date>
        <lido:earliestDate>2002-10-12</lido:earliestDate>
        <lido:latestDate>2002-10-12</lido:latestDate>
        </lido:date>
 </lido:eventDate>
</lido:event>
```

Where the first example has a low precision and the last one is very precise.

Causes:

- Human errors / missing knowledge of cataloguer (e.g. when setting genres)
- Sources for statements too vague / general lack of knowledge (e.g. no research conducted yet)
- Changes of acquisition contest (e.g. different research questions of different projects)
- DATA04.1

Root in the data life cycle: Collect

Identification:

- If applicable: By attributes or statements marking the degree of precision
- REGEX looking for certain expressions marking a varying degree of precision (e.g. “approx.”)

Target state: All data records are available with the same precision.

Preventive improvement:

- Impossible due to the heterogenous precision of available research data for different entities

Retrospective improvement:

- Impossible due to the heterogenous limited precision of known research data

DATA01.6.3 Heterogenous Qualifiers for Uncertainty

Description: To mark a specific degree or type of uncertainty being inherent to an information different qualifiers are used.

Mainly affected quality dimension: Consistency

Other affected quality dimensions: Precision, understandability

Impact on data quality: Using different qualifiers to mark a certain degree or type of uncertainty leads to heterogeneous data and loss of consistency since a user or a script parsing the data cannot be sure what the exact meaning of a qualifier is. Due to the heterogeneity, the data is also less precise because not all cases of a specific level of uncertainty might be retrieved in a query. Furthermore, the semantics of the qualifiers are obscured which leads to a drop of understandability of the data.

Examples:

- “1920?” vs. “1920(?)” vs. “1920*”
- “approx. 1920” vs. “ca. 1920”

Causes:

- No acquisition guide or controlled vocabulary available where qualifiers are defined
- Heterogeneity because different disciplines and their respective customs (“In our domain, ‘*’ is used for...”)
- Changes of acquisition contest (e.g. different research questions of different projects)

Root in the data life cycle: Collect

Identification: REGEX

Target state: If qualifiers are used, it should be in a controlled way, i.e. each qualifier should have own clear semantics, outlined by a definition. Preferably, degrees of uncertainty should be expressed in a structured way via a controlled vocabulary, cf. DATA06.7.1.

Preventive improvement:

- Use of an acquisition guide where qualifiers are explained OR use of a controlled vocabulary for different kinds of uncertainties
- Peer review

Retrospective improvement: Retrieve all cases with qualifiers and manually re-evaluate them. This is time-consuming and eventually, not all cases of qualifiers are discovered (e.g. when a qualifier is used only once or there is a typo).

DATA01.6.4 Heterogeneous Value Representations

Description: In free text fields heterogeneous styles, phrases, etc. are used, which do have equal meanings for describing the same phenomenon.
Mainly affected quality dimension: Usability
Other affected quality dimensions: Precision, consistency, uniqueness, understandability
Impact on data quality:

- Inefficient retrieval: Data records are not able to be found by searching for certain descriptions
- Decrease of comparability and understandability

Examples:

- “im 3. Jhd. n. Chr.” vs. “im 3. Jahrhundert AD” vs. “im dritten Jahrhundert nach Christus”
- “lobby” vs. “foyer”

Causes:

- No arrangements between multiple data acquisitors (e.g. missing guidelines)
- Changes of acquisition contest (e.g. different research questions of different projects)

Root in the data life cycle: Collect
Identification: Synonym phrases in the same data field of different data records
Target state: Specific phenomena are expressed in a consistent style using similar phrases.
Preventive improvement:

- Establish style guidelines on how to record descriptions
- Establish some kind of control (guidelines, controlled syntax/vocabularies if possible)
    - Check guidelines with semi-automatic style checks and/or peer review
    - ↭ DATA10.4

Retrospective improvement:

- Controlled vocabularies: Mapping synonyms
    - Automatic replacement

DATA02 Wrong Data Values

Problems based on faulty data.

DATA02.1 Misspelling

Description: There are mistakes in the spelling of single words.
Mainly affected quality dimension: Understandability
Other affected quality dimensions: Accuracy, uniqueness, usability, trustworthiness
Impact on data quality: Not human-readable or machine-understandable data. In some cases, a misspelling can also alter the meaning of a word completely so that the information content of a record decreases.
Examples:

- Deformed data type literals

Causes: Typing error by cataloguers or overlooked during redaction.
Root in the data life cycle: Collect
Identification: Data field entries do not exist in reference work like dictionaries or lexica.
Target state: All data field entries are spelt correctly. Exceptions are fields, which include deliberately wrong values like inscriptions with mistakes or historical spellings.
Preventive improvement:

- Spellcheck during acquisition.
- If reasonable use controlled vocabulary.

Retrospective improvement: Check all entries using a spell-check with suitable reference works.

DATA02.2 Wrong Information

Description: The information stated in the data do not represent the true scientific facts/information. This mostly happens during the information acquisition by the person entering the data and can generate inconsistencies.

Mainly affected quality dimension: Accuracy

Other affected quality dimensions: Usability, understandability, trustworthiness, timeliness

Impact on data quality: There are wrong statements in the data. Within the database, this can generate inconsistencies while researchers and data users are confronted with errors. This can promote false knowledge and misconceptions, which partially cannot be resolved in the future.

Examples:

- One mistake in the Bildindex took over to cultural literature and even when it got corrected it still prevails in it.

Causes:

- Data sources with wrong information
- Data ageing (data becomes outdated)
- Bad estimations or confusion during acquisition
- Missing accuracy by cataloguers
- The low professional expertise of cataloguers

Root in the data life cycle: Collect

Identification: Information is stated falsely in data. An active search is nearly impossible since data seems to be correct at first glance without expert knowledge about the entity itself.

Target state: All given information is correct.

Preventive improvement:

- Wrong information can partially be prevented by more careful research. For this data acquisitors need more time for each acquisition entity and access to all relevant information.
- Plausibility checks can identify wrong information but can only support

Retrospective improvement:

- Redaction and data curation need to actively search and correct inventory data. This is very time-consuming.
- Plausibility checks can identify wrong information but can only support

DATA02.3 Wrong Use of Controlled Vocabulary / Authority File

Description: A term of a controlled vocabulary or an entity of an authority file is linked incorrectly (i.e. not representing reality) in the data.

Mainly affected quality dimension: Accuracy

Other affected quality dimensions: Consistency, trustworthiness

Impact on data quality:

If a wrong term for a concept (e.g. a genre) is used the data contradicts reality and is incorrect. This may induce users to trust the data (or worse: All data provided) less. Also, this leads to the wrong information being spread.

Examples:

```
<lido:classification>
        <lido:conceptID lido:type="http://terminology.lido-schema.org/identifier_type/uri">
                http://obg.vocnet.org/x001200x
        </lido:conceptID>
        <lido:term>Kupferstich</lido:term>
</lido:classification>
```

where http://obg.vocnet.org/x001200x points to “Schabkunst” instead of “Kupferstich”

Causes:

- Human error (time pressure, inattentiveness, ...)
- Lack of knowledge
- Different rules for cataloguers

Root in the data life cycle: Collect

Identification:

- Manual
- Comparison of basic values to values of the referenced data (when not only the reference is given, but additionally basic information, which should match data of the linked record).

Target state: Every use of a controlled vocabulary and/or authority file should precisely match the knowledge about an entity.

Preventive improvement:

- Staff training
- Peer review before submitting the data
- More time during cataloguing to keep data quality at a high level

Retrospective improvement: Manual inspection and correction of data. This is very time-consuming.

DATA02.4 Misplaced Information

Description: Information is placed in the wrong data field in the data model. Therefore information does not match the field description.

Mainly affected quality dimension: Accuracy

Other affected quality dimensions: Consistency, trustworthiness, uniqueness, understandability

Impact on data quality: The statement made in the data is incorrect as the given value does not correspond to the kind of information expected for this field. Therefore information cannot be found even though they are specified in the data. It is not possible to find out if another field should hold this value or which other value would be correct in this field.

Examples:

- <lido:namePlaceSet>
  <lido:appellationValue>Leonardo da Vince</lido:appellationValue>
  </lido:namePlaceSet>
  where the creator's name has been placed in an element describing a field
- <tei:name>2019-10-10</tei:name> (instead of <tei:date>2019-10-10</tei:date>)

Causes:

- Different rules for cataloguers
- Mistakes during data acquisition (caused by lack of time, knowledge, inattentiveness)
  - Human error (time pressure, inattentiveness, …)
  - Lack of knowledge
  - Data model too complex or ambiguous
- Faulty data transformation
- Incomplete data model: The required or desired information aspects do not fit in any specified field
- Data model too complex or ambiguous
- MODEL02

Root in the data life cycle: Collect

Identification: Data does not match the format (data type, structure, vocabulary) of the field

- Anomaly detection: The format of the statement differs from other statements in this field
- Search manually

Target state: All information is placed in the designated data field.

Preventive improvement:

- Staff training
- Peer review before submitting the data
- More time during cataloguing to keep data quality at a high level
- Automatic format and anomaly tests during data acquisition
- Ensure correctness of the data transformation (e.g. via data transformation testing analogous to model transformation testing)
- Make the data model only as complex as necessary to represent the entities of interest
- Add lists with controlled vocabulary to as many fields as possible

Retrospective improvement:

- Identify misuses of fields via a regular expression, anomaly detection, checking against the vocabulary, patterns
- Manually or semi-automatically (in case you identified a recurring pattern) move the identified misplaced information to the correct field if possible
- Manually add the missing information to the field

DATA02.4.1 Multiple Information in a Single Repeatable Field

Description: Instead of coding data into repeatable fields, all information is put into one field with some delimiters. (Note, that in the case that the field is not repeatable, this is a data model quality problem (MODEL01))
Mainly affected quality dimension: Understandability
Other affected quality dimensions: Accuracy, consistency
Impact on data quality: The information increments are not machine-readable without explicitly defining the delimiter.
Examples:

- <lido:repositoryLocation>München/Munich</lido:repositoryLocation>

Causes:

- Faulty data acquisition

Root in the data life cycle: Plan, collect
Identification: Use of delimiters (“,” , ”;” , “/” , “|”)
Target state:

- When multiple information match in a field, it should be repeatable
- In repeatable fields, the information should never be put into single fields

Preventive improvement:

- Implement repeatable fields, if possible
- Validation: Test input strings wrt. possible delimiters and throw a warning if a delimiter is used

Retrospective improvement:

- Split delimiters in repeatable fields (only semi-automatic to bypass outliers!)

DATA02.5 Inconsistencies based on Dependencies

Problems generated through inconsistencies concerning dependencies between multiple values. Thus, general plausibility rules are violated. These plausibility rules can be viewed as constraints to the data model.

DATA02.5.1 Mismatching Date Dependencies

Description: Because of causality and a linear timeline, some events are related to other events and therefore can only happen at certain time intervals that depend on other datings. The depending dates, however, are outside the logical and possible time intervals.
Mainly affected quality dimension: Consistency
Other affected quality dimensions: Accuracy, precision, uniqueness
Impact on data quality:

- Contradicting information
- DATA01.6

Examples:

- Date of death before the birth date
- Artist's creative period is stated outside the artist's lifetime
- Time of an object's creation is outside its creator's lifetime
- Purchase date before creation date
- Purchase date outside the lifetime of the buyer
- Date before the development of technique (e.g. photography earlier than 1826)

Causes:

- Human error
- Contradictory literature sources
- No plausibility check on data input

Root in the data life cycle: Collect
Identification: Comparison-Pattern (Values)
Target state: Dates stated in the data record should be plausible according to all accompanying information.
Preventive improvement:

- Help during acquisition: An automatic or manual plausibility check could be done on data generation with a warning message. This needs to be supported by the acquisition software.

Retrospective improvement:

- Fields can be found automatically
- Manual correction of single values
    - Possible huge effort depending on the number of occurrences

DATA02.5.2 Mismatching Functional Dependencies of Categorizations

Description: When fields specify other fields (and therefore contain more specific values) these build a functional dependency. Here inconsistencies can develop easily when values do not fulfil the requirements based on their dependency. This is especially the case for sub-categorizations: Each more specific classification shall always have a specific broader classification.
Mainly affected quality dimension: Consistency
Other affected quality dimensions: Accuracy, precision, uniqueness
Impact on data quality:

- No matching information
- DATA01.6

Examples:

- Genre → classification → term → technique (eg. architecture and photography)
- (profession of artist ↔ genre)
- A church is always classified as architecture

Causes:

- Human error
- No plausibility check on data input
- Unclear/blurred facts
- Confusion about combinations of object genres

Root in the data life cycle: Collect
Identification: Contained-Pattern: Value must be in List of allowed values, dependent on other value
Target state: All information do match consistent
Preventive improvement:

- Help during acquisition: allowing only a limited choice depending during data collection (at best software-technical)

Retrospective improvement:

- The wrong fields can be found automatically by plausibility checks
- Correction hast to be done manually

DATA02.5.3 Mismatching Dependencies of Spatial Statements

Description: The specified locations in data records are linked to other values, which can lead to contradictions.
Mainly affected quality dimension: Consistency
Other affected quality dimensions: Accuracy, precision, uniqueness
Impact on data quality:

- Mismatching information
- DATA01.6

Examples:

- Object: Place of origin + Time of origin ↮ Place of artist
- Place of part-object ↮ Place of total object

Causes:

- Human Error
- No plausibility check on data input
- Unclear/blurred facts

Root in the data life cycle: Collect
Identification: Plausibility checks
Target state: All spatial information do match consistently
Preventive improvement: Plausibility check on data input
Retrospective improvement: Manual check on found plausibility violations

DATA02.5.4 Violation of Dependencies between Obligatory Statements (Alternating)

Description: If a condition is fulfilled (e.g. specific fields are (not) filled) in a data record there are mandatory conditions for other fields to be filled.
Mainly affected quality dimension: Completeness
Other affected quality dimensions: Consistency, usability, trustworthiness
Impact on data quality: Missing values at mandatory fields
Examples:

- A specified street could be problematic without specifying the city.
- MIDAS: “5365 (Maßzahl)” without “5364 (Maßart)”

Causes: ignorance, faulty sources
Root in the data life cycle: Collect
Identification: Non-existent pattern (under the condition, that another value is (not) set)
Target state: Mandatory fields are always filled with a value. If a field is mandatory to be set under a fixed condition, this condition is met.
Preventive improvement: The saving of a data record is only possible when all mandatory conditions are fulfilled.
Retrospective improvement: Analyse the database and fill all mandatory but empty fields.

DATA03 Double Information in Data

This section deals with all cases where information appears more than once in a data set.

DATA03.1 Multiple Data Records Describing the Same Entity

Description: There exist at least two data records which describe the same real-life entity. These are hard to identify and to merge. For the identification, a proximity metric is necessary. The merging is especially problematic since (even small) differences in the data sets have to be tested for accuracy to decide which version should be maintained. In the best case, all versions are maintained while all but one version is marked as outdated or false.

Mainly affected quality dimension: Uniqueness

Other affected quality dimensions: Consistency, trustworthiness, completeness

Impact on data quality:

- There can be differences between the data records
    - Hence it is not clear which one is correct
- Updates in data records will only be applied to one, this will lead to contradictions and inconsistencies
- References to one entity may lead to different data records

Examples:

- Two data records with different IDs describing the same painting

Causes:

- Cataloguers are not aware of existing data records
    - Lack of time for research
- Cataloguers are not able to identify that a new record describes the very same entity
- Imports from different databases
- Different views/opinions on content
- The entity was acquired under different research questions which results in a data record for each objective

Root in the data life cycle: Collect

Identification: Compare different data records with a proximity metric.

Target state: Each real-world entity has only a maximum of one associated data record in the database. When data records are merged from multiple sources, all information should be maintained while one record should be marked as the preferred one. False or outdated information should be marked as such. Merged data records should be preserved to maintain resolvability of references and a cross-reference to the preferred record should be made.

Preventive improvement:

- Check existing data records at the creation or import of new data sets.
- Establish a workflow for dealing with duplicates when importing data sets.

Retrospective improvement:

- The reunification of duplicate data records.
    - Identification
    - Merging correctly: Maintaining all information, marking outdated information as such.
- Mark one data record as the preferred one, mark others as outdated and add a reference to the preferred version to them.

DATA03.2 Redundancies in Data

Description: Redundant data: Information is stated in multiple locations in a data record/set.
Mainly affected quality dimension: Consistency
Other affected quality dimensions: Accuracy, consistency, uniqueness, trustworthiness
Impact on data quality:

- Larger data records
- Changes may not be applied to all respective fields equally

Examples:

```
<lido:measurementsSet>
        <lido:measurementType xml:lang="en">width</lido:measurementType>
        <lido:measurementUnit xml:lang="en">mm</lido:measurementUnit>
        <lido:measurementValue>222</lido:measurementValue>
</lido:measurementsSet>
<lido:measurementsSet>
        <lido:measurementType>Breite</lido:measurementType>
        <lido:measurementUnit>mm</lido:measurementUnit>
        <lido:measurementValue>222</lido:measurementValue>
</lido:measurementsSet>
```

Here the unit and the value is redundant.

Causes:

- A field is falsely assigned multiple times
- The same information fits into different fields of the record
- Data model calls for redundancies, e.g. when stating information in different languages
- Lack of time for data acquisitors which leads to inattentiveness

Root in the data life cycle: Collect, assure
Identification: Same content in multiple fields of a single record.
Target state: Information should be stated only once per data record and data set.
Preventive improvement:

- Warning if the same information is registered more than once in a data record
- designing a data model in a way that discourages redundancies

Retrospective improvement:

- Elimination of single redundant fields or restructuration of the record

DATA04 Units

Problems based on units of data values.

DATA04.1 Inconsistent Use of Units or Metric Systems

Description: In a measurement entry units are needed for understandability. Possibly multiple/different units are used depending on the context (DATA01.6). This makes the comparison more complex because conversion is necessary. However, values can get converted to use a different unit. With only a single unit, the data would be comparable without conversion. Therefore it is desirable to use only one metric system and as few units as possible. In the best case, only SI based units are used.

(In the case of only one allowed unit, it can be used as default value. In that special case, no unit has to be stated in the data and Problem DATA04.2 would not apply. Generally, the unit is not specified or a specification is even prevented by the data model.)

Mainly affected quality dimension: Consistency

Other affected quality dimensions: Accuracy

Impact on data quality: Bad comparability

Examples:

- MIDAS “5365 (Maßzahl)” (e.g. “21 cm”, “93,3 g”, “26 mm”, “2,52 m”, “220.000 Liter”, “120,2 t”, “150 kg”, “36 Zoll”)
- “9144 m” instead of “10000 Inch”; 0,9 m instead of 1 Yard

Causes: no specifications during data acquisition

Root in the data life cycle: Collect

Identification: In different field-instances, different units are used

Target state: Only SI base units are used when providing measurements.

Preventive improvement:

- Default unit
- Demand conversion (to SI unit)
- Could be solved by data model design by allowing only one unit of measurement (cm for height) per type and a possibility to add the preferred display unit (m)

Retrospective improvement: Automatically convert the values to use the same measurement system

DATA04.2 Missing Units

Description: A unit puts a number in relation to other numbers. A numeral value without a unit (excluding countings) does not provide useful information.
Mainly affected quality dimension: Understandability
Other affected quality dimensions: Precision, usability, completeness
Impact on data quality: A number, e.g. a measurement without units is not clear. The information is not understandable and hardly usable.
Examples:

- MIDAS “5365 (Maßzahl)” (e.g. “28,5”, “660”, “13,7”)

Causes: employee forgets specification of units during data acquisition
Root in the data life cycle: Collect
Identification: MATCH-Pattern
Target state: Numeric information should always be provided with a unit (if it is not specifying a quantity)
Preventive improvement:

- Specifying a default unit for all field entries and demand conversions if needed
- The additional mandatory field for the unit
- Field bisected for the acquisition form with a mutual obligation
- Warning or error if the entry does not match a regular expression during data acquisition

Retrospective improvement: Search for field entries/element sets/statements without a unit and add them manually

DATA05 References

This section encompasses references to other data records as well as relations of an object to another one which is expressed in the data.

Examples:

- Art ⟶ Artist
- Picture ⇿ Data Record
- Data Record ⟶ Authority Data

DATA05.1 Missing References Between Data Records

Description: A data record describing an entity does not contain references to other records which describe related entities. Thus, relations between entities are not reflected in the data.

Mainly affected quality dimension: Completeness

Other affected quality dimensions: Usability

Impact on data quality: The domain of interest is incompletely represented in the data which results in low data quality. Also, the usability of the records is reduced since the data record is isolated instead of being connected to other data. This impedes making connections to other data (semi-)automatically.

Examples:

- <lido:actor lido:type="http://terminology.lido-schema.org/lido00163">
  <lido:nameActorSet>
  <lido:appellationValue>Botticelli, Sandro</lido:appellationValue>
  </lido:nameActorSet>
  </lido:actor>
  Without providing an URI pointing to an authority file.
- Missing <lido:relatedWork>
- The connection between entities is described in a free text field but not in a machine-readable way

Causes:

- The person editing the data record was not aware of relations to other entities represented in the data set or forgot to add the references
- The person acquiring data does not know how to express a relationship between data records
- The data model does not allow to express the relation
- Acquisition software/technology does not allow to express the relation

Root in the data life cycle: Collect

Identification:

- (semi-) automatically identify empty reference fields (if applicable)

Target state: Any relations between entities which exist and/or are described in the data should also be represented in the data (preferably in a machine-readable way).

Preventive improvement:

- Consider references while creating the data model
- Highlight empty reference fields in data input system before saving the record

Retrospective improvement:

- After identifying empty reference fields manually check if any references can be added
- Modify data model or technology such a way that the required relations can be expressed

DATA05.2 Reference to a Non-Existent Data Record

Description: A record contains a reference to another record in the same or a different data set. However, the referenced record does not exist (anymore). This may also apply to a reference to norm data.

Mainly affected quality dimension: Completeness

Other affected quality dimensions: Usability

Impact on data quality: Information about the related data record is missing. References cannot be resolved and are unclear, thus impeding further use of the still existing record.

Examples:

```
<lido:partOfPlace>
        <lido:placeID
            lido:type="http://terminology.lido-schema.org/identifier_type/uri">http://sws.geonames.org/abcd/</lido:placeID>
        <lido:namePlaceSet>
            <lido:appellationValue>Göttingen</lido:appellationValue>
        </lido:namePlaceSet>
</lido:partOfPlace>
```

where https://www.geonames.org/abcd/ refers to nothing instead of Göttingen

Causes:

- Typo during manual input of the foreign key / ID
- Record has been deleted or moved
- After removal of duplicate records the references have not been updated

Root in the data life cycle: Collect

Identification: Check all references between records regarding the existence of the referenced record or norm data. This should be done on a regular basis, e.g. once a year.

Target state: All records or entries in norm data that are referenced in a data record should exist and be resolvable.

Preventive improvement:

- Automatically check the existence of the referenced record or norm data entry when a reference is stated while creating or updating a new data record. The runtime of this check may harm the efficiency of data editing.
- Remove or (where possible) update references when records are deleted.
- Establish a workflow to regularly check for deleted or moved references records or norm data entries.

Retrospective improvement:

- Remove or (manually) update affected references. References to non-existent data could be detected automatically by a script while the correction requires human intellect and can be very time-consuming when a lot of references are corrupted.

DATA05.3 Reference To Record Not Unique

Description: A record's field which is used for referencing the record from within other records and thus is supposed to function as a unique identifier does not hold unique values across all records in the data set.

Mainly affected quality dimension: Uniqueness

Other affected quality dimensions: Precision, usability

Impact on data quality: References to other records cannot be evaluated unambiguously. This way the reference is not resolvable which leads to a decrease in information content.

Examples:

- MIDAS repeatable field Ob30 relation between object and artist is defined via his/her name (3100) (which is not unique) instead of referencing the (object's or) artist's data record ID
- MIDAS repeatable field Ob35 relation between object and atelier is defined via the atelier name (3600) (which is not unique) instead of referencing the atelier's data record ID
    - Examples for non-unique atelier names found in the MIDAS data: "Hollerbaum und Schmidt", "Adametz, Franz", "Berger, Emerich", "Braune und Schwenke"

Causes:

- A non-unique property of the entity is used for referencing the record
- Inadequate ID management
- Import of records with IDs overlapping those of records already included in the data set
- MODEL10

Root in the data life cycle: Plan, collect

Identification: Check the uniqueness of the values in all identifier fields.

Target state: Other records should be referenced via unique identifiers. For this being the case, records should provide an identifier which is unique in the data set.

Preventive improvement:

- Addition of IDs in the data model for each data record
    - Cf. MODEL10
    - Automatic ID assignment (a combination of integer and institution identifier) for new records
    - Further: Adaptation of IDs of imported records

Retrospective improvement:

- Revise records with ambiguous references manually
- Reconstruct data imports

DATA05.4 Ambiguous Reference to Described (real-life) Entity

Description: The reference to the entity described by the data is ambiguous, i.e. it is not specified which entity the data record describes. Multiple different entities can fall into the scope. In contrast to DATA05.3, the focus here are references to the entities described by the data, not references to other data records.

Mainly affected quality dimension: Uniqueness

Other affected quality dimensions: Precision, usability, understandability, trustworthiness

Impact on data quality: The referenced entity is not identifiable. This leads to a decrease of information content, and the record is probably less usable since its scope is unclear.

Examples:

- MIDAS: Reference to the artist only by name while there may be several artists with the same name

Causes:

- Due to a lack of information, it is unclear which entity is described in the source(s)
- The cataloguing persons have not been aware of similar entities and the ambiguity of this reference.
- The data model lacks options for referencing authority data
- The data model lacks fields for precisely describing the entity

Root in the data life cycle: Collect

Identification:

- Manual test if the real-life entity can be uniquely identified
- Identification may be problematic since there is no way of testing if another object with the same known properties exists.
- Missing references to authority data
- The high number of empty fields

Target state: Each reference to an entity is unambiguous. The entity is identifiable from the reference (value).

Preventive improvement:

- Design data model such that entities can be described precisely and via references to authority data
- Acquire the entity description as exact as possible (for the data record)
    - Add a label with a unique identifier to the real-life entity
    - Reference authority data such as GND
    - Add exact Geo-Coordinates for sedentary entities

Retrospective improvement:

- Manual with great expenditure of time

DATA05.5 Unretrievable Resource from URI Namespaces

Description: URI namespaces need to be chosen with consideration of organizational and technical requirements. Unretrievable resources can lead to information loss and data lacks.

Mainly affected quality dimension: Consistency

Other affected quality dimensions: Reusability, trustworthiness

Impact on data quality: In an open, distributed environment, obsolete URIs are essentially useless data values. A single change in a namespace component can confound the validity of huge numbers of statements in connected datasets. Altering a chosen namespace severely affects the very purpose of a URI. Resolution of changed URIs can become impossible if no redirecting resolver can be installed and maintained under the old namespace. Even where the redirection is possible, the maintenance overhead for the institution in charge of the namespace, and the burden on those having to deal with multiple URIs for a single resource, can be substantial.

Examples:

- Gemeinsame Normdatei (GND)[20]: In 2019, the GND Linked Data Service changed the protocol header in all URIs from http: to https:. Resources under the old URI can still be retrieved, but the new URI replaced the old one as the only "true" identifier. As a consequence, tens of millions of GND URI references that exist throughout the world have essentially become invalid. Any software system not specifically aware of this change will conclude that http://d-nb.info/gnd/7724415-1 is different from https://d-nb.info/gnd/7724415-1. The resulting technical issues are likely to persist for many years to come.
- (more examples needed; e.g. several projects have issued URIs that did not survive the funding period but were nevertheless referenced in other projects)

Causes:

- The namespace is based on a non-persistent DNS[21] domain, often registered for a project and expiring after the end of the funding period. While still a valid identifier, it usually cannot be made resolvable again. Moreover, an expired domain name can come under the control of a non-related (and sometimes embarrassing) owner.
- The registration of the namespace domain expires for some other reason and is taken over by a domain grabber.
- The namespace is based on the domain name of an external contractor and this role is taken over by another contractor, or by the institution itself.
- The namespace insinuates an affiliation with an organization and this turns out to be inappropriate for administrative, political, legal, or business reasons.

Root in the data life cycle: Collect, publish

Identification: Ideally, any URI should represent a retrievable resource. This can be verified by simple network requests. Where resolvability is not feasible or necessary, the namespace domain should at least exist in the Internet DNS, and be under the control of the organization that uses the namespace for creating URIs. This can be verified by querying the DNS or a domain registry service.

Target state: Each reference should be retrievable persistently and standardized by using https or persistent identifiers like DOI[22] or handle.

Preventive improvement:

[20] https://www.dnb.de/EN/gnd

[21] DNS = Domain Name System: https://en.wikipedia.org/wiki/Domain_Name_System

[22] DOI = Digital Object Identifier: ISO 26324:2012(en) Information and documentation — Digital object identifier system https://www.iso.org/obp/ui/#iso:std:iso:26324:ed-1:v1:en

- Due consideration should be given to the longevity of a URI namespace. This applies not only to the DNS domain name, but also to any other URI component such as the protocol prefix, port number, directory path, and fragment identifier as far as these are part of the namespace identifier. Query strings (following a question mark sign) should never be used as URI components.
- Where long-term persistence cannot be guaranteed, the use of services such as purl.org or a URN resolver should be considered from the outset.

Retrospective improvement: Once an existing URI namespace has been changed, the necessary action depends on how - and to what extent - corresponding URIs are already in use. As long as it can be assured that no external party (cooperating institution, aggregator, etc.) uses any URI from the former namespace, the changeover can be performed locally. In all other cases, a (sometimes complex) strategy for mitigating the unintended consequences needs to be developed.

DATA06 Uncertainty

Some facts cannot be known for sure due to a lack of knowledge. Because of likelihoods, inferences, hints, a few data points, etc. the set of possible values can be limited or an assumption can be made.

We consider DATA01.1 as one type of uncertainty.

DATA06.1 Doubtful Data

Description: Statements in the data are unreliable in the sense that their semantic correctness is in question. Due to a lack of clear evidence (or ageing of the data), there is doubt involved.
Mainly affected quality dimension: Trustworthiness
Other affected quality dimensions: Accuracy
Impact on data quality: Data quality is decreased by doubtful data since the information is potentially wrong. However, data quality is improved by qualification and (approximate) quantification of the data's trustworthiness and by giving indications of source.
Examples:

- <h1:Field Type="5130" Value="niederländisch?"/>
- <h1:Field Type="ob30" Value="Herstellung">
  <h1:Field Type="3100" Value="Cuyp, Aelbert Jacobsz."/>
  <h1:Field Type="3470" Value="zugeschrieben"/>
  <h1:Field Type="3475" Value="Maler"/>
  </h1:Field>
- <h1:Field Type="ob35" Value="Herstellung">
  <h1:Field Type="3600" Value="Tranzparenz, Berlin"/>
  <h1:Field Type="3970" Value="ungesichert"/>
  </h1:Field>

Causes:

- Lack of perfect information during data creation
  - The information is subject to ongoing research and is not yet proven or accepted as definitely correct by the research community
    - There is evidence (e.g. in sources) for a statement to be true with an increased probability, but not with 100% certainty
    - Sources, measurements or clues only implicitly point towards something
    - Multiple measurements have different results
    - Sources are dubious regarding their authenticity or accuracy
      - Reliability of sources is in question
      - Limited trust in sources
    - Only a few sources or measurements available
    - Theories are (still) unproven
    - Interpretation of historical sources
  - The person editing the data record is only certain (or confident) to some extent that the statement s/he makes in the data is true
    - The considered sources
      - Only point into a direction
      - Are questionable regarding their authenticity or correctness
      - Are extremely subjective or biased
      - Are too few to know for certain
    - There exist hints contradicting the statement
    - Research performed only partially due to lack of resources
    - Lack of sources
    - Unproven assumption(s)

        - Providing an estimation (i.e. guess) rather than no statement at all
- Through ageing of the data (dynamics), it is uncertain if the data is still up to date and correct (see DATA07.3 Outdated Data)

Root in the data life cycle: Collect

Identification:

- Institution-specific methods for implicitly or explicitly marking doubtfulness, e.g. by appending "?"
- An imprecise statement could be a hint for the cataloguer not being confident enough to make a more precise statement
- Hints in the data concerning the cataloguer's level of confidence
- Date of last modification is long ago
- No or only a few indications of source

Target state: The data should include as much certain information as is available per the current state of research. Information that is not yet accepted as correct by the research community but has an increased probability of being correct should also be included in the data, marked and qualified appropriately (see DATA06.7). The same holds for information with an increased probability of being correct that the person editing the record was not (yet) able to verify.

Preventive improvement:

- Allow cataloguers to do a comprehensive research
- Design data model
    - Such that indications of the source can be added to each statement (enable the representation of research discourses) (see DATA01.1.4)
    - Such that alternative opinions can be expressed and attached with sources (see DATA06.3)
    - Such that the confidence in a statement can be expressed explicitly (see DATA06.7.1)
    - Such that the name or at least expertise of the person editing the record can be added to each statement (see DATA01.1.5)

Retrospective improvement:

- Review doubtful data regularly: Do the latest developments in research brace or falsify the statements in the data?

DATA06.2 Imprecision

Description: Statements in the data do not satisfy the context-specific requirements concerning precision. Thus, they are too imprecise for context-specific use cases. There is a lack of detail or a coarse granularity. The statement implies multiple possible values on a continuum of which at maximum one is correct. These values are all coherent according to some similarity measure. They differ only quantitatively. The boundaries of the set of values indicated by the statement may be known clearly (e.g. “between 1900 and 1910”) or vaguely (i.e. fuzzy) (e.g. “around 1905”). Often temporal and spatial data is affected by imprecision. However, also verbal statements (e.g. for classifying an object) can be imprecise and thus too general or abstract.

Mainly affected quality dimension: Precision

Other affected quality dimensions: -

Impact on data quality:

- The data is of low quality as it is less precise than expected and required by the users.
- DATA10.5

Examples:

- Acquisition based on an illustration of the object, without access to the actual object
  - Additional uncertainties, estimations, guesses (e.g. material)
- ```
  <h1:Field Type="5060" Value="Datierung">
      <h1:Field Type="5064" Value="gegen 1700"/>
  </h1:Field>
  ```
- ```
  <h1:Field Type="ob30" Value="Herstellung">
      <h1:Field Type="3100" Value="Scorel, Jan van"/>
      <h1:Field Type="3470" Value="Kreis"/>
      <h1:Field Type="3475" Value="Maler"/>
  </h1:Field>
  ```
- ```
  <h1:Field Type="5060" Value="Datierung">
    <h1:Field Type="5064" Value="1451/1475"/>
  </h1:Field>
  ```

Causes:

- Information about the research entity is not exactly but only imprecisely known
  - With the current state of research and with all available sources only imprecise information can be ascertained (e.g. measured)
  - Information is based on estimation (e.g. via comparison to other related entities and information)
  - Limited precision of measurement tools and analysis methods
  - Non-reproducibility of physical measurements
  - Noise in measurements
- The current state of research is not exactly but only imprecisely mapped to the data
  - During editing of the data record precise information cannot be determined (with comparative effort)
  - Only imprecise statements in the considered sources
  - Providing an approximation or guess in order to avoid a missing statement
  - Practice of “certainty rather than precision”: Rather giving an imprecise but certain statement than making a precise statement which may be incorrect
  - Dynamics: Statements lacking further contextual information become imprecise over time
- The technology used or the data model does not allow precise statements in the data record or the data model itself contains imprecision
- Authority data is imprecise and thus causes imprecision in the data referencing it

Root in the data life cycle: Plan, collect

Identification:

- Words hinting at estimations (e.g. “ca.”)
- Number rounded off (e.g. to an integer)
  ```

- The suffix of a number consists of many zeros
- Unit too rough
- An interval is given where a single value is expected
    - Undefined or imprecise interval boundaries (e.g. “after ...”)
- Set of discrete possible but mutually exclusive values
- Abstract, too general terms
- Undefined symbols

Target state: The data should meet the context-specific requirements for precision if the current state of research provides this level of precision. Otherwise, imprecision should be explicitly marked and qualified (DATA06.7).

Preventive improvement:

- Specify requirements concerning precision for each field, e.g. number of decimal places, the precision of date and time, geographic coordinates
- System for editing data records could visualize overall imprecision score of the record

Retrospective improvement:

- Review data regularly: Is the current state of research precisely represented in the data?
- Manually review records that contain indications of imprecision

DATA06.3 Contradiction

Description: There are several contradicting statements included in the data. Thus at maximum one of the statements can be true. They either indicate different alternatives for the same information aspect (i.e. field), such as multiple possible birthdates of an artist or span across multiple different fields and violate a plausibility rule that specifies a constraint over these fields (DATA02.5). In contrast to DATA06.2 the alternatives are not similar. They differ qualitatively not just quantitatively. The contradictions are either introduced by accident or intentionally because no definite statements can be made based on the research by the cataloguer or even according to the current state of research.

Mainly affected quality dimension: Consistency

Other affected quality dimensions: -

Impact on data quality: As the statements contradict each other, only one of them can be correct. Since the user cannot tell which of the given statements is correct, the data quality is decreased.

Examples:

- Providing a specific creator for a painting in the metadata and referencing another person as a creator in a free text description
- ```
  <h1:Field Type="ob30" Value="Herstellung">
    <h1:Field Type="3100" Value="Aertsen, Pieter"/>
    <h1:Field Type="3475" Value="Maler"/>
    <h1:Field Type="ob30rl" Value="oder"/>
  </h1:Field>
  <h1:Field Type="ob30" Value="Herstellung">
    <h1:Field Type="3100" Value="Beuckelaer, Joachim"/>
    <h1:Field Type="3475" Value="Maler"/>
    <h1:Field Type="ob30rl" Value="oder"/>
  </h1:Field>
  ```

Causes:

- Unintended contradiction:
  - Different persons editing the record have different opinions/interpretations/understandings of the entity of interest and (unknowingly) transfer these to (different parts of) the data without indicating the contradiction explicitly
  - When incorrect statements are updated, thus new correct statements are added, the old statements are not removed or marked as outdated
  - Human error: Inattentiveness
  - The person editing the record accidentally makes ambiguous statements based on institution-specific practices (DATA09)
- Intended contradiction:
  - Research debate:
    - Multiple researchers disagree on a specific topic
    - Multiple contradicting opinions/sources are reflected in the data
  - The person editing the record based on his/her research identified multiple possible but mutually exclusive values and could not figure out which of the values is correct so he or she included all of them in the data

Root in the data life cycle: Collect

Identification:

- Check plausibility rules via pattern matching
- Detect violations of typical dependencies (e.g. genre and occupation of the associated artist) identified via statistical analysis (e.g. anomaly detection) or machine learning methods
- Field repetition
- Multiple values in a single field (e.g. separated via special characters, such as “,” or “|”)
  ```

Target state: The data should be free of any contradictions except for those that per the current state of research represent equally valid, contradicting opinions of researchers and those that represent multiple possible but mutually exclusive values identified but not solved by the person editing the record. These contradictions should be expressed explicitly and augmented with contextual information (DATA06.7). The data should be updated as soon as there are new findings regarding the topic of discussion.

Preventive improvement:

- Define plausibility rules for a data set
- Check plausibility rules during data creation
- Include constructs for explicitly expressing mutually exclusive possible values and research debates in the data model
    - Provide different constructs for expressing multiple values per field (e.g. “these two artists together created the artwork”) vs. multiple mutually exclusive possible values per field (e.g. “the artwork was either created by artist A or by artist B”) as currently both meanings are often expressed in the same or a very similar way which causes ambiguous data
    - Provide different constructs for saying that another but the still unknown possibility may exist
    - Provide context information for each possible value, especially indication of the source
- Include constructs for explicitly marking outdated statements in the data model

Retrospective improvement:

- Semi-automatically detect contradictions as described above and resolve them manually or semi-automatically in case of uncertain statements that contradict a plausibility rule

DATA06.4 Unmarked Uncertainties in Data

Description: Statements in the data are uncertain (e.g. imprecise, contradicting, unreliable or incomplete), but are not marked as such.
Mainly affected quality dimension: Trustworthiness
Other affected quality dimensions: Accuracy
Impact on data quality: The data users cannot recognize uncertainties in the data and thus are not able to handle the affected information with caution when using them as a basis for their further scientific work. For example, an uncertainty score per record cannot be calculated. Strictly speaking, statements suggesting they are certain when they are incorrect. Missing indication of uncertainties also prevents formalization of the semantics of uncertainties and thus also (semi-) automatic reduction of uncertainty.
Examples:

- Stating a statue is made in 1700 whereas no exact source for this statement is available and thus the statement is uncertain (i.e. imprecise or a guess)

Causes:

- No guidelines telling the cataloguers how to mark uncertainty
- The data model does not allow for marking uncertainty
- Person or institution does not want to lose reputation
- Stems from the uncertainty of the person editing the record or from the current state of research
- It is not specified which type of information is expected for a specific field (e.g. single value or interval). If e.g. in such a case an interval is given, the user does not know whether this is a hint for imprecision (interval vs. unknown single value in the interval).

Root in the data life cycle: Plan, collect
Identification:

- Hint: Inconsistencies
- Hint: Incompleteness
- Abstract (verbal) statements
- Imprecise statements hint at estimation

Target state: Any uncertain statements in the data should be marked and qualified (context) appropriately.
Preventive improvement:

- Teach cataloguers to mark uncertainties
- Make uncertainties explicit in the data model
    - Distinguish between different kinds of uncertainty
- Design data model such that it allows indications of source
- The data model should contain separate fields for specifying interval boundaries
- Specify which type of information is expected per field in the data model

Retrospective improvement:

- Manually review inconsistencies, imprecise and abstract statements

DATA06.5 Implicitly Marked Uncertainties

Description: Statements in the data are uncertain, but only implicitly marked as such. Thus, they are, for example, marked via additional characters included in the same field instead of being specified via additional appropriate constructs. Implicit encodings of uncertainty are institution-specific and often ambiguous. Thus these encodings, in general, are not understandable. (cf. DATA01.6.3)

Mainly affected quality dimension: Trustworthiness

Other affected quality dimensions: Precision

Impact on data quality: Both machines and humans cannot easily understand and assess the uncertainty of the information. This means information about uncertainty, cannot easily be queried and appropriately presented to the user. For example, an overall uncertainty score per data record cannot easily be calculated. Implicit marking also prevents the formalization of the semantics of uncertainties and thus also (semi-)automatic reduction of uncertainty.

Examples:

- ```
  <h1:Field Type="5060" Value="Datierung">
      <h1:Field Type="5064" Value="um 1910 ?"/>
  </h1:Field>
  ```

Causes:

- The data model or technology does not allow to explicitly express uncertainty (MODEL01)
- Cataloguer does not understand the modelling concepts for expressing uncertainty

Root in the data life cycle: Plan, collect

Identification:

- Patterns: Structural patterns, regular expressions
- Data mining: Anomaly detection

Target state: Any uncertain statements in the data should be expressed explicitly. To achieve this, the underlying data model has to allow for expressing any types of uncertain statements.

Preventive improvement:

- Design data model such that different forms of uncertainty can be expressed explicitly
- Strict syntax rules per field to prevent unwanted implicit encodings of uncertainty

Retrospective improvement:

- Improve data model and transfer old data to a new model after detection of implicitly encoded uncertainties via patterns and manual review by domain experts
  ```

DATA06.6 Dependency Between Uncertain Statements Not Expressed

Description: There exists a dependency between multiple uncertain statements in the data, but the dependency is not (explicitly) specified in the data. This often occurs if multiple possible but mutually exclusive values are given for a field A and another field's (field B) value is dependent on which of these values is correct for field A.

Mainly affected quality dimension: Accuracy

Other affected quality dimensions: Understandability, completeness

Impact on data quality: Since dependencies between statements in the data are not (explicitly) specified, data users are not aware of these dependencies. Thus, users erroneously consider combinations of statements which actually contradict the dependency rules and thus are incorrect as possibly correct.

Example:

- MIDAS data record "00000001": The value in field 2996 depends on the value in field 2864
-
```
<h1:Field Type="ob28" Value="zeitweiliger Verwalter">
    <h1:Field Type="2864" Value="Marburg / Wiesbaden"/>
    <h1:Field Type="2900" Value="Central Collecting Point"/>
    <h1:Field Type="2996" Value="1945.05/1946.08 / 1945.06/1948.08"/>
</h1:Field>
```

Causes:

- The data model does not allow expressing dependencies between uncertain statements
- Cataloguer forgets to add dependency information
- System for editing records does not allow adding dependencies in a simple and understandable way

Root in the data life cycle: Collect

Identification:

- Manually check if there exists an unmarked dependency between multiple uncertain statements of a data record or a set of related records

Target state: Any dependency between uncertain statements should be expressed explicitly in the data.

Preventive improvement:

- Modify the data model such that any dependencies can be expressed
- Provide an intuitive way for entering dependencies in the UI of the system for editing records

Retrospective improvement:

- Manually add missing dependencies

DATA06.7 Missing Qualification of Uncertainty

Description: There are uncertain statements in the data which are not further qualified. Therefore, they lack contextual meta-information regarding the uncertainty, such as the type of uncertainty, the reason for uncertainty (e.g. subjective vs. objective, lack of resources, etc.), the level of confidence (DATA06.7.1), indication of source, the expertise of the person editing the record, duration of time the record was edited, date of last modification etc.
Mainly affected quality dimension: Trustworthiness
Other affected quality dimensions: -
Impact on data quality: If context information for uncertain statements is missing from the data, users cannot assess the consequences of using uncertain data as e.g. they cannot assess the level of trustworthiness or whether they might find certain information elsewhere.
Examples:

- <h1:Field Type="5130" Value="niederländisch?"/>

Causes:

- The data model does not allow to express context information for uncertain statements
- Cataloguer does not enter context information for uncertain statements

Root in the data life cycle: Plan, collect, describe
Identification:

- Check if the data model supports context information for uncertain statements
- Find empty fields that should hold context information for uncertain statements

Target state: Any uncertain statements should be accompanied by information specifying the context of uncertainty, i.e. the cause for and level of uncertainty. Regarding the cause of uncertainty we first of all need to distinguish between subjective and objective uncertainty (i.e. the *source* of uncertainty). The former is present if the cataloguer him or herself is not certain about a statement s/he makes in the data. Objective uncertainty, in contrast, means that according to the current state of research no certain statement can be made by any researcher in the field. Furthermore, the *cause* for subjective or objective uncertainty should be specified more precisely via controlled vocabulary. For example, subjective uncertainty may be caused by a lack of time for doing research when creating a data record. The *level* of uncertainty (or confidence) should be expressed via controlled vocabulary (DATA06.7.1).
Preventive improvement:

- Design data model such that uncertain statements can be augmented with meta-information about the uncertainty
    - Source
    - Cause
    - Level

Retrospective improvement:

- Manually revise uncertain statements (e.g. found via pattern-based methods) and add contextual information

DATA06.7.1 Degree of Uncertainty Not Specified

Description: Some statements are more certain than others but this is not expressed in the data.
Mainly affected quality dimension: Accuracy
Other affected quality dimensions: Precision, understandability, trustworthiness
Impact on data quality: When statements have a different degree of certainty but this is not stated in the data, statements that are less certain can appear to be more certain and vice versa. This way, the data suggests (un)certainty where there is none. Thus the data is not correct (in a narrower sense) nor is it precise and may lead to wrong deductions in reuse. It further prevents the calculation of the overall level of uncertainty per record.
Examples:

- Hypotheses with strong vs. weak evidence are placed next to each other without differentiation

Causes:

- The data model does not allow for stating different degrees of uncertainty
- Staff cannot determine the proper level of uncertainty

Root in the data life cycle: Collect
Identification: Lexical analysis of the data fields if different degrees of uncertainty are indicated. Otherwise, cases can only be identified by checking the data manually.
Target state: If a piece of information is not certain this should be marked and classified according to the degree of uncertainty. The degree of uncertainty should preferably be taken from a controlled vocabulary.
Preventive improvement:

- Provide a controlled vocabulary for different gradations of uncertainty
- Staff training concerning said controlled vocabulary
- Support evaluation of uncertainty degrees with software, e.g. implementing a drop-down for selecting the proper value from the vocabulary in the acquisition software

Retrospective improvement: To achieve this, a controlled vocabulary has to be developed or reused and all fields that are possibly affected have to be reviewed manually - if the statement's source is clear. Otherwise, sources have to be determined first. This is very time-consuming, the one way or the other.

DATA06.8 Heterogeneous Representations of Uncertainty

Description: Occurrences of a specific type of uncertainty are expressed in multiple different (i.e. heterogeneous) ways in the data. This affects both implicit markings via qualifiers (DATA01.6.3) and explicit representations.

Mainly affected quality dimension: Consistency

Other affected quality dimensions: Understandability, simplicity, uniqueness

Impact on data quality: Data quality is negatively impacted by heterogeneous representations of a specific type of uncertainty since they must all be taken into account and be processed when humans or machines interpret (or analyse) the data.

Examples:

- MIDAS - DDK Practices for Uncertainty not uniform
    - Alternative possible values are sometimes listed together in one field whereas in other cases the field is repeated for each value

Causes:

- MODEL04
- MODEL09
- DATA01.6.3

Root in the data life cycle: Plan

Identification:

-

Target state: For each type of uncertainty there should exist exactly one uniform way to express it. This should be an explicit representation.

Preventive improvement:

- Conduct a requirements analysis concerning the types of uncertainty that should be expressed within the data model
- Model required information explicitly and uniquely

Retrospective improvement:

- Improve data model such that each type of uncertainty is modelled explicitly and uniquely
- Transfer existing data into the new model via pattern-based transformations

DATA07 Dynamics

All inventory research data are changeable and never finished because there can always be new knowledge discoveries. Therefore the existing data can become outdated and needs to be changed.

DATA07.1 Data Dynamics not Documented (in the Data itself)

Description: Due to new information the values of a data field can change. Oftentimes older states of knowledge are simply deleted and replaced with more recent information which results in information loss about former data. Sometimes, however, outdated information is saved in free text fields.

Mainly affected quality dimension: Trustworthiness

Other affected quality dimensions: Completeness

Impact on data quality: Former states of knowledge are lost which is a loss of information content. Changes are not transparent to the user which can lead to a severe drop in the data provider's trustworthiness.

Examples:

- https://sammlungen.uni-goettingen.de/lidoresolver?id=record_kuniweb_365457
- https://sammlungen.uni-goettingen.de/objekt/record_kuniweb_365457/1/-/
- https://hdl.handle.net/21.11107/d0c7a32e-2652-479d-883f-5b51e3fc4bb4

  <lido:descriptiveNoteValue>
  [...]Bisher als "Vielbrüstige" bezeichnet. Heutige Deutung: Der Göttin Artemis galt der Stier als heilig, er war ihr Begleiter. [...]</lido:descriptiveNoteValue>[23]

Causes:

- Missing change management
- Data history not expressible in the data model

Root in the data life cycle: Collect

Identification: Searching for certain phrases in free text fields. In the case of deleted data, nothing can be done with reasonable effort.

Target state: Changes in data should be addressed explicitly, e.g. as a changelog (cf. TEI). Alternatively, data fields with outdated information could provide information about their up-to-dateness and a date stamp when the field has been marked as outdated. In XML-based data formats, this could be achieved by attributes (e.g. "currentness='out-of-date' currentness-change='2019-01-01' ", whereas in graph-based formats two triples/statements could be added (e.g. "X" "hasCurrentnessStatus" "outdated" and "X" "isOutdatedSince" "2019-01-01").

Preventive improvement:

- Offer standardized mechanisms that allow for providing information on former data, e.g. as separate input field in GUI
- Choosing/updating data models: Should be able to track changes in data
- The software automatically adds an entry in a changelog when the input of a file in an already saved record changes

The first improvements can be rather costly since the schema has been changed which also results in software changes.

Retrospective improvement: This problem can only be tackled if the data model allows for attributes/statements that describe data changes. Even so, improvement may not be feasible:

[23] Homepage of University collection Göttingen: https://hdl.handle.net/21.11107/d0c7a32e-2652-479d-883f-5b51e3fc4bb4 (accessed on 08/04/2020) https://sammlungen.uni-goettingen.de/lidoresolver?id=record_kuniweb_365457 (accessed on 08/04/2020)

Deleted data is lost, and information in free text fields might not be found automatically–depending on the degree of standardization of object descriptions.

DATA07.2 Model Dynamics not Documented in Data

Description: When the data model changes, the data structure can change too or be taken over while not using the new features. However, no hint indicates the model change.

Mainly affected quality dimension: Trustworthiness

Other affected quality dimensions: Understandability, completeness

Impact on data quality: Missing information on model changes is not transparent to the user and might lead to problems in applications using the data. Furthermore, the missing information can be viewed as a lack of information content.

Examples: An element or predicate is renamed in the data model and the data records are updated without any note of it in the data.

Causes:

- The data model does not provide the possibility to provide information on changes due to model updates (MODEL05)
- Human error: Documentation has been forgotten

Root in the data life cycle: Plan, collect

Identification: No elements/statements exist where information about changes in the data model are stored.

Target state: All data records have a revision element/statement wrt. data model updates and resulting changes in the data records.

Preventive improvement:

- Provide an element/statement for changes in the data model
- Develop workflows for change management, containing e.g. a script that automatically adds an element/statement about data model updates and their consequences to the record

Retrospective improvement:

- Update the data model and provide an element/statement for changes in it
- If applicable, write and run a script that automatically adds an element/statement about past data model updates and their consequences to the record. At least the most recent change (which might be the addition of revision statements) of the data model should be made transparent.

DATA07.3 Outdated Data

Description: Obsolete data has not been edited lately. Since updates regarding the entity of interest have not been transferred to the record describing the entity, it is possible that the record does not represent the entity correctly anymore. This also applies to cases in which former uncertain information which is represented in the data over time is solidified or proven to be (in)correct. In this case, the record also needs to be updated accordingly. The required update frequency depends on the kind of entity represented by the data and its volatility.

Mainly affected quality dimension: Timeliness

Other affected quality dimensions: Accuracy

Impact on data quality: Outdated data causes low data quality since the data may not correctly represent the current state of research regarding the described entity. Additionally: If users cannot assess how old the data is (e.g. due to missing information concerning timestamp), it decreases their trust in the accuracy of the data.

Examples:

- Former assumptions that turned out to be false are not marked as such in the data
- The date of creation for artwork is now more precisely known than before because of scientific progress but this increase in precision is not reflected in the data
- lido:repositorySetComplexType[@type = “current”] where a user cannot be sure if this still holds

Causes:

- The research knowledge about entities is growing, the assigned data records have to be updated, when new knowledge is gained
- Not enough resources for keeping data up to date
- Changes or updatable information haven not been noticed
- No existing workflow for keeping data up to date

Root in the data life cycle: Plan, preserve

Identification:

- Hint: The last update of a record lies further in the past than desired for the specific task
- Hint: Low frequency of data record change

Target state: Each record should describe the current state of research regarding the described entity. As soon as the entity changes or new information is available, the record should be updated.

Preventive improvement:

- Review records regularly
    - Problems: Not enough resources and possible no access to the latest information
- Develop a workflow to update records, e.g. for the case that an object is lent another institution
- Identify elements/attributes/statements that are prone to change (frequently) and regularly screen them

Retrospective improvement:

- Review (outdated) records at least each time a new record is created which references the outdated one

DATA08 Subjectivity

Description: In the data, subjective opinion is presented, i.e. judging descriptions which are heavily shaped by an opinion. This can occur in fields with free text like descriptions.
Mainly affected quality dimension: Usability
Other affected quality dimensions: Trustworthiness, completeness (other/objective opinion)
Impact on data quality: The delivered knowledge is shaped by opinion and therefore not objective.
Examples:

- “beautiful”, “ugly”, “good”, “bad”, “fantastic”, “cruel”

Causes:

- Bad data acquisition

Root in the data life cycle: Collect
Identification:

- Finding judgmental formulations using Containment-Pattern

Target state: The information is collected objectively.
Preventive improvement: training of cataloguers
Retrospective improvement:

- Rework data records with descriptions containing judging formulations.

DATA09 Implicit Knowledge

Description: Knowledge about the context of data collection is not explicitly expressed in data and to understand the delivered knowledge completely, interpretations and background knowledge are necessary.

Mainly affected quality dimension: Understandability

Other affected quality dimensions: Uniqueness, completeness, accuracy

Impact on data quality: Data becomes more trustworthy for others if implicit knowledge is (at least partly) communicated. Assessment of completeness of data is difficult. Data is only partially fit for reuse.

Examples:

- Foto Marburg holds negatives within its collection that need to be documented. Therefore for each negative one record contains a certain signature with numbers and characters. The character stands for the format of the negative, e.g. LA = 35mm format, B = medium format. So the information on the format is implicit within this information.
- Context of data collection, e.g. within a certain research project is only partially derivable from the object number
- Researchers of a project are not interested in names of bishops within a historical source. For data review, it is important to know that s/he left out bishops' names within the person's vocabulary.
- Data is generated only by and for use of one researcher within a project.

Causes:

- For local use of data not everything needs to be made explicit because people generating and using data have implicit knowledge (of the place, collection, scientific context etc.)
- Human error: Making implicit knowledge explicit has been forgotten
- Lack of time/resources
- Some information is taken as self-evident and therefore not expressed explicitly

Root in the data life cycle: Collect

Identification:

- Review of Workflow- or Project documentation
- Data that is not explicitly collected can hardly be identified. A qualitative review by a domain expert might give an idea of what is not expressed within the data.

Target state:

- The knowledge that is important for the understanding of data is made explicit.
- Context, background and origin of the data generated are made explicit.
- Datasets of projects are published with Metadata about the collection process and context.

Preventive improvement:

- Exact documentation of which data is generated and which is not
- Develop a data management plan
- Analyze the data model and its usage instructions exactly for reusability

Retrospective improvement:

- Note context of generation/ source within the metadata for datasets

DATA09.1 Not Standardized Symbols are used to express certain Facts

Description: During the acquisition, cataloguers make up symbols to ease their work effort or to communicate with the reviewing person. Free text fields are used for this. The meaning of these symbols is rarely documented and may even change over time so that it is possible that their meaning is not unique. Often they represent uncertainties towards the described fact. They are usually not known to persons outside the workflow of data collection, so the meaning or content of these symbols stays hidden in other stages of the data life cycle. This results in confusion of users in later stages when looking upon the data. Also, the information stays implicit and is not machine-readable. In retrieval systems, information can hardly be found.

Mainly affected quality dimension: Uniqueness

Other affected quality dimensions: Understandability, Trustworthiness, Timeliness, Accessibility, Completeness

Impact on data quality: Values art not unique concerning their semantic meaning. Data is only human-readable and might be outdated. Since information is not explicit it is not findable via retrieval systems.

Examples:

- The field for Comments is used to discuss questions by marking them with “Q:”
- MIDAS
    - “x”, “y”, “?” = “unknown, please add later”

Causes:

- Incomplete data model: Data does not fit in any field
    - Missing fields to express uncertainties or relations between fields
- Missing documentation of local data-collection practices
- Missing possibilities to communicate within workflow other than via data-fields
- Missing or few quality checks

Root in the data life cycle: Collect

Identification: If symbols are known, data fields can be checked for them.

Target state:

- Data model and cataloguing software provide machine-readable possibilities to express uncertainties.
- Communication about Work in Progress is done outside the data and cataloguing system

Preventive improvement:

- Communication of work in progress e.g. need for inherent quality check by a supervisor can be done via ticketing-system.

Retrospective improvement: Check data for used symbols to manually review and update information.

DATA10 Controlled Vocabulary

DATA10.1 Violation of Controlled Vocabularies (Use of Custom Values)

Description: When using controlled vocabulary in specific fields, the registered value has to be in a list of allowed values matching the field specification to ensure comparability and uniformity. When the intended vocabulary is ignored, new entries are not uniform or comparable. This might be the case, when new values are used, a vocabulary contains homonymous values, the use of synonymous values, which are not included in the vocabulary or misspellings. In some cases, the new values automatically get added to the controlled vocabulary.

Mainly affected quality dimension: Consistency

Other affected quality dimensions: Accuracy, usability, understandability, trustworthiness, completeness

Impact on data quality: Comparability across data records suffers because various terms (preferred/alternative/custom) are used to describe the same entity.

Examples:

- "cup" is intended, but the synonymous value "mug" is used (use of alternative term instead of the preferred term)
- MIDAS-Vocabularies can be freely supplemented: E.g. the vocabulary did contain "Druck" and "„Kupferstich (Druck)" for "Sachbegriff 5230".
- "watch" is contained as a clock in the vocabulary, but used in the meaning of "guard"

Causes:

- The *acquisition software* allows for both free text and controlled terms
- Misspelling or ignorance of rules in data generation or revision
- Multiple individuums use the same vocabulary developing their own rules for entries each, i.e. not using the vocabulary correctly
    - Description for controlled terms unclear
- Human error: Use of alternative terms instead of preferred terms
- In LIDO files: The data is not validated with the schema file

Root in the data life cycle: Collect

Identification:

- Contained-Pattern: Value must be in the list of allowed values, dependent on other value
- Wrong use of homonymous values need extensive (manual?) contextual analysis

Target state: All fields/elements/statements that should be controlled use a term of one of the mandatory controlled vocabulary/vocabularies.

Preventive improvement:

- Acquisition program can/should forbid free text in fields which should be controlled
- Train of staff wrt. the correct usage of controlled vocabularies
- Establish workflow for dealing with unclear definitions of terms (local rules, contributing to vocabulary, ...)
- Use URI of vocabulary term instead

Retrospective improvement:

- Manually revise fields with values which are not part of the vocabulary
- The cases that are examined manually can be reduced by excluding fields with links to a controlled vocabulary.

DATA10.2 Missing Reference to Authority Data (Global Comparability)

Description: The data lacks references to authority data. Authority data provides unique identifiers for unambiguously referencing a specific entity and includes all relevant information about this entity.

Mainly affected quality dimension: Uniqueness

Other affected quality dimensions: -

Impact on data quality: References to entities such as persons or places without references to corresponding authority records may be ambiguous as e.g. a person's name is not unique. Furthermore, it is not possible to reuse information provided by authority data without a proper reference (e.g. as a URI).

Examples:

- Examples for authority data:
  - IconClass for Iconography
  - AAT for Art and Architecture
  - GND for Persons, Corporate Bodies and Subject Headings
  - TGN for geographical places
- <lido:actorID>123456789</lido:actorID>

  instead of

  ```
  <lido:actorID lido:type="http://terminology.lido-schema.org/identifier_type/uri">
      http://d-nb.info/gnd/123468493
  </lido:actorID>
  ```

Causes:

- Authority data needed does not exist
- Technology or data model does not allow to add a reference to authority data
- The person editing the data record does not add the reference to authority data
- Vocabulary is imprecise (DATA10.5)

Root in the data life cycle: Collect

Identification:

- Fields for referencing authority data are empty

Target state: Data should include as many references to authority data as possible. Ideally, each entity mentioned should be linked to an apt authority file.

Preventive improvement:

- Make fields for referencing authority data mandatory where you expect it to exist
  - Problem: What if the authority data does not exist?
- Check input on data creation and warn the user if no URI to authority file is present

Retrospective improvement:

- Modify technology or data model such that all possible references to authority data can be added
- (Semi-) automatically try to find corresponding authority records
  - Automatically suggest possible mappings
  - Let the domain expert decide

DATA10.3 Unattended Vocabularies or Thesauri

Description: It is necessary to maintain vocabularies and thesauri. This includes to correct, update and complete the associated databases. Furthermore, there should be a fixed concept of what the vocabulary is about.

Mainly affected quality dimension: Usability

Other affected quality dimensions: Accuracy, precision, consistency, uniqueness, understandability

Impact on data quality:

- Connections between data values are missing.
- Synonymous data is not identifiable.
- Links to incorrect terms decrease quality
- Retrieval gets worse (recall and precision)

Examples:

- "church" does not contain a reference to "building"
- "mug" is not entered as a synonym for "cup"
- "chappel" is missing in the vocabulary, but used in the data
- In MIDAS, a copper engraving (Kupferstich) should be described via Sachbegriff „5230: Druck" plus the naming of „5300: Kupferstich" as technique. „Kupferstich (Druck)" has been added as an entry in the vocabulary 5230, also "Kupferstich (Technik)" is to be found in the vocabulary 5300 for techniques. -> two existing ways to express the same information

Causes:

- No active tending of the vocabulary
- No automatic alert when entering values
- Concept of the vocabulary changes / gets fuzzy over time
- Several institutions sharing one vocabulary

Root in the data life cycle: Plan, collect, assure, analyze

Identification:

- Terms that are used in a record's field/element/statement/ not present in the defined vocabulary
- The vocabulary contains gaps

Target state:

- The vocabularies are up to date and maintained.
- All used values are fully specified (definition, preferred/alternative, narrower/broader term, ...)
- All data fields in the vocabulary are specified
- Synonyms are marked such, i.e. as an alternative term

Preventive improvement:

- Warning during data collection when entering a value which is not in the vocabulary
- Assign a person as curator for the Vocabulary

Retrospective improvement:

- Manually complete vocabulary entries
- Manually merge synonymous values or mark as "alternative/ USE" in LIDO
- Add missing values which are used in the data to the vocabulary.

DATA10.4 Unnecessary Use of Custom Controlled Vocabulary

Description: The controlled vocabulary used is a custom made one (e.g. specific for an institution), although all terms the custom vocabulary encompasses are already represented in a wide-spread and established domain-specific controlled vocabulary.

Mainly affected quality dimension: Reusability

Other affected quality dimensions: -

Impact on data quality: Using custom made controlled vocabularies while established ones are already at hand significantly decreases the interoperability and, thus, the reusability of data records. A local URI or term has to be mapped to existing vocabularies which impedes applications using data from more than one source.

Examples:

- <lido:conceptID lido:type="http://terminology.lido-schema.org/identifier_type/uri">http://some-local-museum.org/Kupferstich</lido:conceptID>

  instead of

  <lido:conceptID lido:type="http://terminology.lido-schema.org/identifier_type/uri">http://vocab.getty.edu/aat/300041341http://vocab.getty.edu/aat/300041341</lido:conceptID>

Causes:

- Lack of knowledge about existing and established controlled vocabularies
- Lack of knowledge about how to reference established controlled vocabularies
- Result of DATA01.6.4

Root in the data life cycle: Collect

Identification:

- REGEX for URIs which test if the URI matches a well-known vocabulary

Target state: All references to a controlled vocabulary should refer to public, well-known and established controlled vocabularies (if possible). The use (and thus the development) of custom controlled vocabularies should be restricted to cases where no apt term in a wide-spread existing vocabulary is at hand.

Preventive improvement:

- Training on how to deal with controlled vocabularies
- Training on existing and established domain-specific vocabularies
- Writing manuals/docs about good LOD practice

Retrospective improvement:

- Re-map existing local references to more wide-spread options (if possible). Cases could be identified by REGEX or a simple database query. More frequent occurrences of terms (e.g. “colour slides” in a photo library) could be replaced automatically while less prominent terms have to be handled manually.

DATA10.5 Imprecise Controlled Vocabulary

Description: The controlled vocabulary does not satisfy the context-specific requirements concerning precision. This occurs if controlled vocabulary is referenced by other data which is required to be more precise. Often temporal and spatial data is affected.

Mainly affected quality dimension: Precision

Other affected quality dimensions:

Impact on data quality:

- Imprecision within a controlled vocabulary either leads to unwanted imprecision in other data which references the controlled vocabulary (DATA06.2) or to the omission of references to controlled vocabulary (DATA10.2).

Examples:

- Authority data only provides information on big cities (e.g. Mannheim) but not on suburbs.
- Controlled vocabulary provides information for material “paper” but does not allow to differentiate in detail e.g. between “transparent paper” and “tracing paper”.

Causes:

- A controlled vocabulary is referenced in use cases that are different from those it was designed and intended for. More suitable authority data is missing.
- As for values from continuous domains the level of granularity can, in theory, be refined endlessly, a controlled vocabulary can never be precise enough for all kinds of use cases.

Root in the data life cycle: Plan, collect, integrate

Identification:

- Locate imprecise data and check manually whether it is caused by an imprecise controlled vocabulary
- Missing references to a controlled vocabulary

Target state: A controlled vocabulary should satisfy the context-specific requirements concerning precision.

Preventive improvement:

- Choose a controlled vocabulary that suits best the requirements concerning the precision
- Do research or review to add or edit vocabulary entries meeting the requirements concerning the precision
- Design data model such that references to a controlled vocabulary which is imprecise can be explicitly marked as such

Retrospective improvement:

- For missing controlled vocabulary references or imprecision caused by an imprecise controlled vocabulary check manually whether meanwhile, a suitable and precise controlled vocabulary exists
- Do research or review to add or edit vocabulary entries to meet the requirements concerning the precision

DATA10.6 Incomplete Controlled Vocabulary

Description: The controlled vocabulary is too restrictive. Thus, it is incomplete concerning the terms needed to describe the phenomena at hand.

Mainly affected quality dimension: Completeness

Other affected quality dimensions: Correctness, precision

Impact on data quality: If the controlled vocabulary does not contain a term for making a correct statement about the entity of interest, we can either make no statement at all or give an improper term and thus make an incorrect statement. Hence, data quality is decreased since the data does not describe the entity as correct and close as it could if the controlled vocabulary was extended.

Examples:

- The source of a statement can only be stated through a reference to literature but not through e.g. a historical photograph or the cataloguer's expertise

Causes:

- Controlled vocabulary grows over time instead of being fully designed for all use cases before being employed
- New use cases (e.g. new projects) appear after the controlled vocabulary has been created

Root in the data life cycle: Plan

Identification:

- Controlled fields that are left empty

Target state: The controlled vocabulary should contain all terms necessary to make correct statements about the entity in all intended use cases.

Preventive improvement:

- Precisely specify all use cases before designing the controlled vocabulary as far as possible
- Use existing vocabularies and do research or review to add or edit vocabulary entries with regard to the requirements

Retrospective improvement:

- Extend or edit controlled vocabulary by additional words and phrases if necessary
    - (semi-) automatically check whether old data needs to be updated

DATA11 Violation of Formal Specifications

Description: Some fields have concrete syntactical specifications, which have to apply to all fields of a kind. Invalid data in those fields does violate these specifications.

Mainly affected quality dimension: Consistency

Other affected quality dimensions: Consistency, uniqueness, usability, understandability, trustworthiness, completeness

Impact on data quality:

- The data are not explicitly readable, neither from employees nor automatically.

Examples:

- Postcode, street names
- Numbers (binary, decimal, hexadecimal) (Letters, Signs)
- Datings have a fixed format (dd/mm/yyyy; mm/dd/yyyy; yyyy-mm-dd ...)
- No long text in classifying terms (terms, classification)

Causes:

- Human error
- No format check on data acquisition

Root in the data life cycle: Collect

Identification: MATCH-Pattern (regular expression)

Target state: All fields do match their syntactical specifications.

Preventive improvement: Check in the data acquisition form

Retrospective improvement: Manually revise fields with values which do not match the requirements

DATA12 Incompatible Data Types

Description: A field in the database is specified to have a specific non-primitive or primitive data type (e.g. boolean, integer, float), but the registered value or instance does not fulfil the specifications of the data type in the data model.

Mainly affected quality dimension: Consistency

Other affected quality dimensions: Correctness, precision, usability

Impact on data quality: The entries are not machine-understandable and maybe even incomprehensible or ambiguous for users of data.

Examples:

- Float value instead of integer
- String value instead of boolean

Causes:

- Wrong inputs during data collection
- No (automatic) data validation concerning data model specification (during data acquisition)

Root in the data life cycle: Collect

Identification:

- MATCH-Pattern: The registered value does not fulfil the required structure of the data type
- Validation of the data model

Target state: All registered values match the structure of the data type which is associated with its data field.

Preventive improvement: Verification during data acquisition: Check if the registered value corresponds to the expected data type, e.g. by validating the input against a schema.

Retrospective improvement: Manually correct all data entries which do not match the data type.

4. Quality Problems Concerning Data Transformations

As data transformation, we define the conversion of data structured by a certain data model into the same or another varying data model. We like to differentiate data transformation from the model transformation, meaning the change of a certain data model concerning his size, structure or format. This section encompasses all quality problems that stem from transforming data from one data model to another.

TRANS01 Losses (/Reduction) During Data Transformations

Description: During a transformation, information out of the source model is not considered or the source structure of the data record cannot be recreated clearly. Losses can occur on contextual data (the target model does not receive single data values) or structural data (the dependencies between data fields represented in the structural form in the source model). However, sometimes losses during transformations are wanted e.g. due to legal regulations (licenses) concerning the restriction of information transfer.

Mainly affected quality dimension: Completeness/correctness

Other affected quality dimensions: Consistency, reusability

Impact on data quality:

- Information loss of data values or dependencies represented by the structure

Examples:

- Information of single fields are not preserved
- Structural information is ignored (e.g. licenses or financial information)

Causes:

- Institutional restrictions like legal or financial information
- Bad transformation definition
- Model incompatibilities
- Incomplete or incorrect mapping

Root in the data life cycle: Plan, collect, submit, preserve, integrate, publish

Identification: It is not possible to recreate the exact initial source record. Some information is not represented in the target data record.

Preventive improvement:

- Evaluate the definition of the transformation / test the implementation

Retrospective improvement:

- Identify data loss by comparison and complete the data sets
- Add the missing information into the transformed data
- Supplement the definition of the transformation and regenerate the output data.

TRANS02 Incorrect Mapping During Transformation

Description: During a data transformation, information represented in the source model is incorrectly mapped to the target model. Thus, the information is mapped to model elements that are not intended to hold this kind of information.

Mainly affected quality dimension: Correctness

Other affected quality dimensions: -

Impact on data quality: Transformation quality is decreased through incorrect mapping since the transformation produces data of low quality. That is the case since the information that is captured in the wrong model elements decreases correctness and understandability of the data. The actual meaning of the wrongly mapped statements can only be reconstructed by reversing the transformation if the transformation is injective.

Examples:

- <tei:persName> is mapped to <lido:namePlaceSet> (where the former denotes a person's name and the latter a place's name)

Causes:

- Human error: The person specifying the transformation does not have a fully correct understanding of the data models (maybe because the model is too complex, compare MODEL06)
- Unspecific field concept (MODEL02) in the source model: A too generally specified field in the source model is directly mapped to a field in the target model which is intended to hold only a subset of the kind of information that the field in the source model can hold

Root in the data life cycle: Preserve

Identification:

- Technique based on model transformation testing (see [24])
    - Execute the transformation with representative test data and check the correctness of the result via oracle functions
    - Check correctness either by comparing the actual output data with the expected output data or by validating contracts which specify expected properties of the output data

Target state: The data transformation should translate the data which is present in the source model to the target model. The resulting data must conform to the target model specification.

Preventive improvement:

- Transformation writers must have a full understanding of both models
- Check if the target model supports all information that is supported by the source model: Do not map this unsupported information
- Be careful with unspecific field concepts: If a field in the source model is intended to hold different kinds of information, the field in the target model the information is mapped to may depend on the kind of information that is mapped
- Test-driven development: Design test cases before specifying the transformation

Retrospective improvement:

- Model transformation testing: Identify wrongly mapped information and the involved model elements
- Adapt transformation specification appropriately

[24] Selim, Gehan M. K., James R. Cordy, and Juergen Dingel. „Model transformation testing: the state of the art". ACM Press. 2012. 10.1145/2432497.2432502

TRANS03 Change of Too Much Data Records During a Mass Change

Description: When many data records are changed to correct a certain value, oftentimes the chosen scope is too extensive or there are outlier records (which should not get the change). While the intended records are getting improved, the additional data records lose information or even generate inconsistencies like contradictions.

Mainly affected quality dimension: Correctness

Other affected quality dimensions: Consistency, precision

Impact on data quality:

- DATA01.6
- Information loss
- DATA02.2

Examples:

- In DDK in the past, they changed a whole number range by overwriting the dates with a more general time period. Hereby multiple correct and concrete dates got lost.

Causes:

- Changing large amounts of data. (Projects, ID-ranges, data records with specific characteristics)
- Light-headed adding records to the scope. (Outliers)

Root in the data life cycle: Assure

Identification:

- In the selected scope of data records, there are records.

Target state:

- Only data records that are safely getting improved by the intended change are changed by a mass change.

Preventive improvement:

- Excessively test the scope for undesired data records.
- Versioning: Secure a version of the data before the changes.

Retrospective improvement: Impossible, lost information is not recoverable.