Evaluating Data Quality for Integration of Data Sources

John Krogstie

Norwegian University of Science and Technology (NTNU),
Sem Sælandsvei 7-9, N-7030 Trondheim, Norway
krogstie@idi.ntnu.no

Abstract. Data can be looked upon as a type of model (on the instance level), as illustrated e.g., in the product models in CAD and PLM-systems. In this paper we use a specialization of a general framework for assessing quality of models to be able to evaluate the combined quality of data for the purpose of investigating potential challenges when doing data integration across different sources. A practical application of the framework from assessing the potential quality of different data sources to be used together in a collaborative work environment is used for illustrating the usefulness of the framework for this purpose. An assessment of specifically relevant knowledge sources (including the characteristics of the tools used for accessing the data) has been done. This has indicated opportunities, but also challenges when trying to integrate data from different data sources typically used by people in different roles in an organization.

Keywords: Product modelling, data integration, data quality.

1 Introduction

Data quality has for a long time been an established area [2]. A related area that was established in the nineties is quality of models (in particular quality of conceptual data models) [21]. Traditionally, one has here looked at model quality for models on the M1 (type) level (to use the model-levels found in e.g., MOF [4]). On the other hand, it is clear especially in product and enterprise modeling that there are models on the instance level (M0), an area described as containing data (or objects in MOF-terminology). Thus our hypothesis is that also data quality can be looked upon relative to more generic frameworks for quality of models. Integrating data sources is often incorrectly regarded as a technical problem that can be solved by the IT-professionals themselves without involvement from the business side. This widespread misconception focus only on the data syntax and ignores the semantic, pragmatic, social and other aspects of the data being integrated that can lead to costly business problems further on.

Discussions on data quality must be looked upon in concert with discussions on data model (or schema) quality. Comprehensive and generic frameworks for evaluating modelling approaches have been developed [13, 19, 23], but these can easily

J. Grabis et al. (Eds.): PoEM 2013, LNBIP 165, pp. 39–53, 2013.

become too general for practical use. Inspired by [22], suggesting the need for an inheritance hierarchy of quality frameworks, we have earlier provided a specialization of the generic SEQUAL framework [13] for the evaluation of the quality of data and their accompanying data models [16]. Whereas the framework used here is the same as in [16], the application of the framework for looking at quality aspects when integrating data sources is novel to this paper.

In section 2, we present the problem area and case study for data integration. Section 3 provides a brief overview of SEQUAL, specialized for data quality assessment. An example of action research on the case, using the framework in practice is provided in section 4. In section 5, we conclude, summarizing the experiences applying the SEQUAL specialization.

2 Description of the Problem Area of the Case-Study

LinkedDesign[1] is an ongoing international project that aims to boost the productivity of engineers by providing an integrated, holistic view on data, actors and processes across the full product lifecycle. To achieve this there is a need to evaluate the appropriateness of a selected number of existing data sources, to be used as a basis for the support of collaborative engineering in a Virtual Obeya [1]. Obeya – Japanese for ”large room” – is a term used in connection with project work in industry, where one attempted to collect all relevant information from the different disciplines involved in the same physical room. Realizing a Virtual Obeya means to provide a “room” with similar properties, which is not a physical room, but exists only on the net.

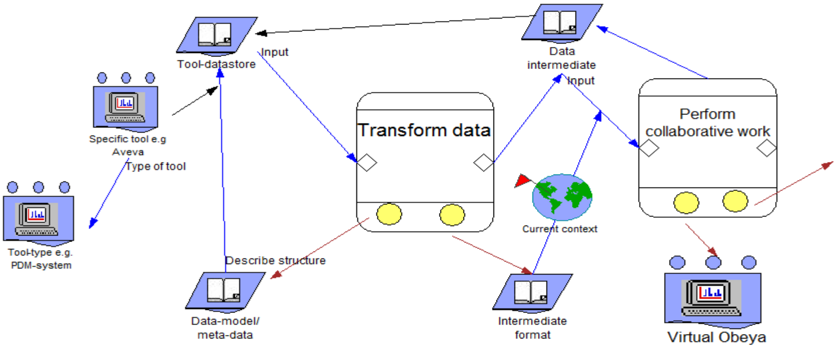

Fig. 1. Approach to knowledge access and creation in a Virtual Obeya [1]

The selected data sources are of the types found particularly relevant in the use cases of the project. When we look at *quality* of a data source (e.g., a PDM tool), we look on both the structure of the stored data in the left of Fig 1. (the data model, including meta-data) and the characteristics of the data itself, in light of our goal for

[1] www.linkeddesign.eu/

reuse and revisualization of data, in a way that might be annotated and/or updated through use. The users are meant to perform collaborative work using the Virtual Obeya. The Obeya presents context specific information based on the persons involved in the collaboration and other relevant information on products, projects, locations, tasks, tools, rules and guidelines etc. The data is mediated from existing work tools and is transformed depending on the context. The data presented and worked on in the Virtual Obeya can be annotated with other context-oriented information that potentially is stored for future use.

3 Introduction to Framework for Data Quality Assessment

SEQUAL [13] is a framework for assessing and understanding the quality of models and modelling languages. It has earlier been used for evaluation of modelling and modelling languages of a large number of perspectives, including data [15], object [11], process [14, 26], enterprise [17], and goal-oriented [10, 12] modelling. Quality has been defined referring to the correspondence between statements belonging to the following sets:

- ***G,*** the set of goals of the modelling task.
- ***L,*** the language extension.
- ***D***, the domain, i.e., the set of all statements that can be stated about the situation. Domains can be divided into two parts, exemplified by looking at a software requirements specification model:
  - Everything the computerized information system is supposed to do. This is termed the *primary domain*.
  - Constraints on the model because of earlier baselined models. This is termed the *modelling context.* In relation to data quality, the underlying data model is part of the modelling context.
- ***M***, the externalized model itself.
- ***K***, the explicit knowledge that the audience have of the domain.
- ***I***, the social actor interpretation of the model
- ***T***, the technical actor interpretation of the model

The main quality types are:

- Physical quality: The basic quality goal is that the externalized model ***M*** is available to the relevant actors (and not others) for interpretation (***I*** and ***T***).
- Empirical quality deals with comprehensibility of the model ***M***.
- Syntactic quality is the correspondence between the model ***M*** and the language extension ***L***.
- Semantic quality is the correspondence between the model ***M*** and the domain ***D***.
- Perceived semantic quality is the similar correspondence between the social actor interpretation ***I*** of a model ***M*** and his or hers current knowledge ***K*** of domain ***D***.
- Pragmatic quality is the correspondence between the model ***M*** and the actor interpretation ***(I*** and ***T)*** of it. Thus whereas empirical quality focus on if the model is

understandable according to some objective measure that has been discovered empirically in e.g., cognitive science, we at this level look on to what extend the model has actually been understood.
- The goal defined for social quality is agreement among actor's interpretations.
- The deontic quality of the model relates to that all statements in the model ***M*** contribute to fulfilling the goals of modelling ***G***, and that all the goals of modelling ***G*** are addressed through the model ***M***.

When we structure different aspects according to these levels, one will find that there might be conflicts between the levels (e.g., what is good for semantic quality might be bad for pragmatic quality and vice versa). This will also be the case when structuring aspects of data quality. We here discuss means within each quality level, positioning the areas that are specified by *Batini* et al. [2], Price et al. [24, 25] and *Moody* [21]. Points from these previously described in [16] are emphasised using italic.

3.1 Physical Data Quality

Aspects of persistence, data being *accessible* (Price) for all (*accessibility* (Batini)), *currency* (Batini) and *security* (Price) cover aspects on the physical level. This area can be looked upon relative to measures of persistence, currency, security and availability that apply also to all other types of models. Tool functionality in connection with physical quality is based on traditional database-functionality.

3.2 Empirical Data Quality

This is addressed by *understandable (Price).* Since data can be presented in many different ways, this relates to how the data is presented and visualized. How to best present different data depends on the underlying data-type. There are a number of generic guidelines within data visualization and related areas that can be applied. For computer-output specifically, many of the principles and tools used for improving human computer interfaces are relevant at the empirical level.

3.3 Syntactic Data Quality

From the generic SEQUAL framework we have that there is one main syntactic quality characteristics, **syntactical correctness**, meaning that all statements in the model are according to the syntax and vocabulary of the language.

Syntax errors are of two kinds:

- **Syntactic invalidity**, in which words not part of the language are used.
- **Syntactic incompleteness**, in which one lack constructs or information to obey the language's grammar.

Conforming to metadata (Price) including that the data conform to the expected data type of the data (as described in the data model) are part of syntactic data quality.

This will typically be related to syntactic invalidity when e.g., the data is of the wrong data-type.

3.4 Semantic Data Quality

When looking upon semantic data quality relative to the primary domain of modelling, we have the following properties:

Completeness in SEQUAL is covered by *completeness (Batini), mapped completely (Price)*, and *mapped unambiguously (Price).*

Validity in SEQUAL is covered by *accuracy (Batini)*, both syntactic and semantic accuracy as Batini has defined it, the difference between these is rather to decide on how incorrect the data is, *phenomena mapped correctly (Price), properties mapped correctly (Price)* and *properties mapped meaningfully (Price).* Since the rules of representation are formally given, *consistency (Batini)/mapped consistently (Price)* is also related to validity. The use of meta-data such as the source of the data is an important mean to support validity of the data.

Properties related to the model context are related to the adherence of the data to the data model. One would expect for instance that

- All tables of the data model should include tuples
- Data is according to the constraints defined in the data-model

The possibility of ensuring high semantic quality of the data is closely related to the semantic quality of the underlying data model. When looking upon semantic quality of the data model relative to the primary domain of modelling, we have the following properties: *Completeness (Moody and Batini)* (number of missing requirements) and *integrity (Moody)* (number of missing business rules).

Completeness (*Moody*) (number of superfluous requirements) and *integrity (Moody)* (number of incorrect business rules) relates to validity. The same applies to Batini's points on *correctness with respect to model* and *correctness with respect to requirements.*

3.5 Pragmatic Data Quality

Pragmatic quality relates to the comprehension of the model by the participants. Two aspects can be distinguished:

- That the interpretation by human stakeholders of the data is correct relative to what is meant to be expressed.
- That the tool interpretation is correct relative to what is meant to be expressed.

Starting with the human comprehension part, pragmatic quality on this level is the correspondence between the data and the audience's interpretation of it.

The main aspect at this level is *interpretability* (Batini), that data is *suitably presented* (Price) and data being *flexibly presented* (Price). Allowing *access to relevant metadata* (Price) is an important mean to achieve comprehension.

3.6 Social Data Quality

The goal defined for social quality is *agreement.* The area *quality of information source (Batini)* touches important mean for the social quality of the data, since a high quality source will increase the probability of agreement.

In some cases one need to combine different data sources. This consists of combing the data-models, and then transferring the data from the two sources into the new schema. Techniques for schema integration [5] are specifically relevant for this area.

3.7 Deontic Data Quality

Aspects on this level relates to the goals of having the data in the first place. Aspects to decide *volatility (Batini)* and *timeliness (Batini)/ timely (Price)* needs to relate to the goal of having and distributing the data. The same is the case for *type-sufficient (Price),* the inclusion of all the types of information important for its use.

4 Application of the Framework

Looking at the sets of SEQUAL in the light of the case of the LinkedDesign project, we have the following:

- *G:* There are goals on two levels. The goal to be achieved when using the base tool and the goal of supporting collaborative work using data from this tool as one of several sources of knowledge to be combined in the Virtual Obeya. Our focus in the case is on this second goal.
- *L:* The language is the way data is encoded (e.g., using some standard), and the language for describing the data model/meta-model.
- *M:* Again on two levels, the data itself and the data-model.
- *A:* Actors i.e., the people in different roles using the models, with a specific focus on the collaborators in the use-cases of the project.
- *K:* The relevant explicit knowledge of the actors (A) in these roles
- *T:* Relates to the possibilities of the languages used to provide tool-support in handling the data (in the base tools, and in the Virtual Obeya)
- *I:* Relates to how easy it is for the different actors to interpret the data as it can be presented (in the base tool, and also in a Virtual Obeya)
- *D:* Domain: The domain can on a general level be looked upon relative to the concepts of an upper-level ontology. We focus on perspectives captured in the generic EKA - Enterprise Knowledge Architecture of Active Knowledge Models (AKM) since these have shown to be useful for context-based user interface development in other projects [19, chapter 5]. Thus we look on information on: Products, tasks, goals and rules (from standards to design rules), roles (including organizational structure and persons, and their capabilities) and tools.

Based on this we can describe the quality of data more precisely for this case:

- **Physical quality** relates to:
  - If the data is available in a physical format (and in different versions when relevant) so that it can be reused in the Virtual Obeya.

  - Possibility to store relevant meta-data e.g., on context
  - Availability of data for update or annotation/extension in the user interface
  - Availability of data from other tools
  - Data only available for those that should have access in case of there being security aspects
- **Empirical quality** is not directly relevant when evaluating the data-sources per se. Guidelines for this is relevant when we look upon how data can be presented in tools (and in the Virtual Obeya).
- **Syntactic quality**. Are the data represented in a way following the defined syntax including standards for the area?
- **Semantic quality**. Do the data sources potentially contain the expected type of data? Note that we here look on the possibility of representing the relevant types of data, obviously the level of completeness is dependent on what is represented in the concrete case. Tools might also have mechanisms for supporting the rapid development of complete models.
- **Pragmatic quality**. Is data of such a type that it can be easily understood (or visualized in a way that can be easily understood) by the stakeholders.
- **Social quality**. Is there agreement on the quality of the data among the stakeholders? Since different data comes from different tools, and often need to be integrated in the Virtual Obeya, agreement on interpretation of data and of the quality of the data sources among the involved stakeholders can be important.
- **Deontic quality**: Shall we with the help of data from the data source be able to achieve the goals of the project? Whereas the treatment at the other levels is meant to be generic, we have here the possibility to address the particular goals of the case explicitly. An important aspect of the case is to reduce waste in lean engineering processes [20]. In LinkedDesign, the use case-partners and other project partners have prioritized the waste areas, and we have used this input to come up with the following list of waste to be avoided as the most important:
  - Searching: time spent searching for information
  - Under-communication: Excessive or not enough time spent in communication
  - Misunderstanding:
  - Interpreting: time spent on interpreting communication or artifacts
  - Waiting: delays due to reviews, approvals etc.
  - Extra processing: excessive creation of artifacts or information

4.1 Evaluations of Relevant Tool-Types

In this project, based on the needs of the use cases, we have focused on the following concrete tools and tool types in the assessment.

- Office automation: Excel
- Computer-Aided Design (CAD): PDMS, Autocad, Catia V5
- Knowledge-based Engineering (KBE): KBEdesign
- Product Lifecycle Management (PLM/ PDM): Teamcenter, Enovia
- Enterprise Research Planning (ERP): SAP ERP (R/3), MS Dynamics

Not all the case organizations used all tool-types. We here focus on one of the organizations which had a need for integration of Excel-data, KBE and PLM-data. In the following we present the treatment of these areas.

4.2 Quality of Excel Data

Much data and information relevant for engineers and other business professionals is developed and resides in office automation tools like Excel [7].

Features Supporting Physical Quality of Excel Data. Data in tools like Excel can be saved both in the native format (.xls, .xlsx), in open standards such as .html, .xps, .dif, and .csv-files, and in open document formats (e.g., .ods), thus Excel-data can be made available in well-established forms following de jure and de facto standards, and thus can be easily made available for visualization and further use. One can also export e.g., PDF-versions of spreadsheets for making the information available without any possibility for interaction. Ensuring secure access to the data when exported is only manually enforced. Since the format is known, it is possible to save (updated) data from e.g., a Virtual Obeya, feeding this back to the original spreadsheet.

Features Supporting Empirical Quality of Excel Data. Excel has several mechanisms for data-visualizations in graphs and diagrams to ensure nice-looking visualizations and these visualizations can be made available externally for other tools. The underlying rules and macros in the spreadsheets are typically not visualized.

Features Supporting Syntactic Quality of Excel Data. Although the syntax of the storage-formats for Excel is well-defined, and standard data-types can be specified, there is no explicit information on the category of data (e.g., if the data represents product information). (Calculation) rules can be programmed, but these are undefined (in the formal meaning of the word), and the rules are in many export formats (such as .csv) not included.

Features Supporting Semantic Quality of Excel Data. You can represent knowledge of all the listed categories in a spreadsheet, but since the data-model is implicit, it is not possible to know what kind of data you have available without support from the human developer of the data, or by having this represented in some other way.

Features Supporting Pragmatic Quality of Excel Data. As indicated under empirical quality you can present data in spreadsheets visually, which can be shared (and you can potentially update the visualization directly), but as discussed under semantic quality, one do not have explicit knowledge of the category of the data represented.

Features Supporting Social Quality of Excel Data. Since Excel (and other office automation tools) typically are personal tools (and adapted to personal needs, even in cases where a company-wide template has been the starting point), there is a large risk that there are inconsistencies between data (and the underlying data model) in different spreadsheets and between data found in spreadsheets and in other tools.

Features Supporting Deontic Quality of Excel Data. Where much engineering knowledge is found in spreadsheets, it can be important to be able to include this in aggregated view in a Virtual Obeya. On the other hand, an explicit meta-model for the data matching a common ontology must typically be made in each case, thus it can be costly to ensure that all relevant data is available. As long as you keep to the same (implicit) meta-model for the data in the spreadsheet, you can update the data in the Virtual Obeya and have it transferred to the original data source. On the other hand, if you need to annotate the data with new categories it is not easy to update the spreadsheet without also updating the explicit meta-model without manual intervention.

Looking upon the waste forms we have the following

- Searching: When Excel is used, there is often data in a number of different Excel-sheets developed by a number of different people, and it is hard to know that one have the right version available.
- Under-communication: There is no explicit data-model, thus the interpretation of data might be based on labels only, which can be interpreted differently by different persons. A number of (calculation) rules are typically captured in Excel-sheets without being apparent.
- Misunderstanding: Due to potential different interpretation of terms, misunderstandings are likely.
- Interpreting: Since the meaning of data is under-communicating, the time to interpret might be quite long.
- Waiting: If data must be manually transformed to another format to be usable this might be an issue.
- Extra processing: Due to the versatility of tools like Excel, it is very easy to represent additional data and rules, even if they are not deemed useful by the organization.

4.3 Quality of Data in KBE Tools

KBE - Knowledge based engineering has its roots in applying AI techniques (especially LISP-based) on engineering problems. In [18], four approaches/programming languages are described: IDL, GDL, AML, and Intent!, all being extensions of LISP. In LinkedDesign, one particular KBE tool is used; KBEdesign™. The KBeDesign™ is an engineering automation tool developed for Oil & Gas offshore platform engineering design and construction, built on top of a commercial Knowledge Based Engineering (KBE) application (Technosofts AML), being similar to the AML sketcher. In the use case, there are two important data sources: The representation of the engineering artifacts themselves, and the way the engineering rules are represented (in AML) as part of the code.

Features Supporting Physical Quality of KBE Data. Knowledge and data is hard-coded in the AML framework. There exists classes for exporting the AML code into XML (or similar), however some information might be lost in this process. There are also classes for querying the AML code for the information you want, along with classes for automatic report creation. It is possible in KBEdesign to interact with most systems in principle. What is so far implemented is import/export routines to analysis software like GeniE, STAAD.Pro. Drawings can be exported to DWG (AutoCAD format). When the model is held within the tool, access rights can be controlled, but it is hard to enforce this when the model is exchanged to other tools. There is limited support for controlling versions both in the rule-set and in the models developed based on the rule-set. As for the rules, these are part of the overall code which can be versioned. Some rules related to model hierarchy and metadata (not geometry) for export to CAD and PLM systems are stored in a database and can be set up per project. Some capability to import data contained in the CAD-system PDMS is implemented.

Features Supporting Empirical Quality of KBE Data. Geometric data can be visualized as one instantiation of a model with certain input parameters. There are also multiple classes for different kind of finite element analysis of the model. Whereas the engineering artifact worked on is visualized in the work-tool, the AML-rules are not available for the engineer in a visual format. For those developing and maintaining the rule-base, these are represented in a code-format (i.e., structured text).

Features Supporting Syntactic Quality of KBE Data. In AML, datatypes are not defined. Programs might run even with syntax errors in formulas as there are both default values, and other mechanisms in place to ensure that systems can run with blank values. The data is stored in a proprietary XML-format, although as indicated it is also possible to make the model available using CAD-standards, but then only the information necessary for visualization is available. Options are available within AML for import and export to industry-standard file formats, including IGES, STEP, STL, and DXF. New STEP standards going beyond the current standards for CAD-tools that are interesting in connection to KBE codification are:

- The standard for construction history that is used to transfer the procedure used to construct the shape, referred to as ISO 10303-55.
- Standards for parameterization and constraints for explicit geometric product models, providing an indication of what are permissible to change refer to ISO 10303-108 for single parts and ISO 10303-109 for assemblies.
- Standard for what is known as 'design features', refer to ISO 10303-111.

Features Supporting Semantic Quality of KBE Data. The focus in KBEDesign is the representation of product data. AML is used to represent engineering rules. There are also possibilities in the core technology to represent process information related to the products. Note that an OO-framework has some well-known limitations in representing rules, e.g., for representing rules spanning many classes [8]. The AML framework also supports dependency tracking, so that if a value or rule is updated, everything that uses that value or rule is also changed. Dynamic instantiation is supported, providing potential short turnaround for changes to the rule-set.

Features Supporting Pragmatic Quality of KBE Data. The experiences from the use case indicate that it is very important to be able to provide rule visualizations, and that these can be annotated with meta-data and additional information. Standard classes in the AML framework allow you to query AML models, generating reports. Data can be visualized any way you want in AML, and if the required visualization is not part of the standard AML framework, then it can be created. It is practical to have everything working in the same environment, but it can be difficult for non-experienced users to find the right functionality.

Features Supporting Social Quality of KBE Data. KBE is a particular solution for engineering knowledge, and experiences from the use case indicate that there is not always agreement on the rules represented. The KBEDesign tool is used for developing oil-platform-designs, but for other engineering and design tasks, other tools are used. Export to tools used company-wide such as PDMS is important to establish agreement, and thus, social quality of the models.

Features Supporting Deontic Quality of KBE Data. An important aspect with object-oriented, rule-based approaches is the potential for supporting reuse across domains. Summarizing relative to factors for waste reduction in lean engineering

- Searching: Representing all rules in the KBE-system is useful in this regard, but they are to a limited degree structured e.g., relative to how rules influence each other, which rules are there to follow a certain standard etc.
- Under-communication: Since AML-rules are accessible as code only, it can be hard to understand why different design decisions are enforced.
- Misunderstanding: Can result from not having access to the rules directly;
- Interpreting: Additional time might be needed for interpretation for the above mentioned reason
- Waiting: If not getting support quickly for updating rules (if necessary), this can be an issue. The use of dynamic instantiation described under semantic quality can alleviate this, on the other hand one needs people with specific coding skills to add or change rules;
- Extra processing: Might need to represent rules differently to be useful in new situations. On the other hand if using the abstraction mechanism in a good way, this can be addressed.

4.4 Quality of Data in PDM/PLM Tools

Product lifecycle management (PLM) is the process of managing the entire lifecycle of a product from its conception, through design and manufacture, to service and disposal. Whereas CAD systems focus primarily on early phases of design, PLM attempts to take a full lifecycle view. PLM intends to integrate people, data, processes and business systems and provides a product information backbone for companies and their extended enterprise. There are a number of different PDM/PLM-tools. Some tools that were previously CAD tools like Catia have extended the functionality to become PLM-tools. The following is particularly based on literature review and interview with representatives for Teamcenter, which according to Gartner group is the market leader internationally for PLM tools. There is typically a core group of people creating information for such tools, and a vast group of people consuming this information.

Features Supporting Physical Quality of PDM/PLM Data. Core product data is held in an internal database supported by a common data model. The data can be under revision/version and security (access) control. Some data related to the product might be held in external files e.g., office documents. There can also be integration to CAD tools and ERP-tools (both ways). For Teamcenter for instance, there is CAD-integration (with Autocad, Autodesk, SolidWorks, Unigraphics, I-deas NX, Solid Edge, Catia V5, Pro Engineer) and ERP-integration (bi-directional with SAP ERP (R/3), MS Dynamics and Oracle). In addition to access on workstation, it is also possible to access the data on mobile platforms such as iPAD. Data can also be shared with e.g., suppliers supporting secure data access across an extended enterprise. This kind of functionality should also make it easier to support the access of data in the PLM-system from outside (e.g., also from a Virtual Obeya). Teamcenter have multisite functionality, but it does not work well to work towards the same database over long distances.

Features Supporting Empirical Quality of PDM/PLM Data. PLM tools typically support 2D and 3D visualization of the products within the tool. These are typically made in CAD tools. CAD tools typically have good functionality to visualize the product data in 3D. Because of its economic importance, CAD has been a major driving force for research in computational geometry and computer graphics and thus for algorithms for visualizations that one typically focus on as means under the area of empirical quality.

Features Supporting Syntactic Quality of PDM/PLM Data. Storage of PLM-data is typically done according to existing standards. PLM XML is supported in Teamcenter, in addition to the formats needed for export to CAD and ERP tools mentioned under physical quality.

Features Supporting Semantic Quality of PDM/PLM Data. As the name implies, the main data kept in PLM systems is product data, including data relevant for the process the product undergoes through its lifecycle. Schedule information and workflow modeling is supported in tools such as Teamcenter, but similar to CAD tools, the function of the parts in the product is not represented in most tools. Compliance management modules can support representation of regulations (as a sort of rules).

Features Supporting Pragmatic Quality of PDM/PLM Data. Relevant context information can be added to the product description supporting understanding. PLM systems have become very complex and as such more difficult to use and comprehend. The size of the products (number of parts) has also increased over the years. Whereas a jet engine in the 1960s had 3000 parts, in 2010 it might have 200000 parts. Reporting is traditionally in Excel, but newer tools can support running reports on the 3D-model, presenting the results as annotation to this. The Teamcenter tool has been reported to be hard to learn if you are not an engineer.

Features Supporting Social Quality of PDM/PLM Data. PLM systems are systems for integrating the enterprise. When implementing PLM-systems one needs to agree on the system set-up, data-coding etc. across the organization. Thus when these kinds of systems are successfully implemented, one can expect there to be high agreement on the data found in the tool in the organization. Note that a similar issue that is found in ERP systems, the so-called work and benefit disparity might occur (this problem was originally described in connection to so-called groupware systems [6]). Company-wide application often require additional work from individuals who do not perceive a direct benefit from the use of the application. When e.g., creating new parts, a large number of attributes need to be added, thus it takes longer time to enter product-information in the beginning.

Features Supporting Deontic Quality of PDM/PLM Data. Looking upon the waste forms we conclude the following

- Searching: Large models and a lot of extra data might make it difficult to get an overview and find all the (and only the) relevant information. On the other hand, since one have a common data-model, it should be easier to find all the data relevant for a given product.
- Under-communication: Since extra data has to be added up front for the use later in the product life cycle, it is a danger that not all necessary data is added (or is added with poor quality), which can lead to the next two issues:
- Misunderstanding: Can be a result of under-communication.
- Interpreting: When engineers and other groups need to communicate, one should also be aware of possible misunderstandings, given that it seems to be hard to learn these tools if you are not an engineer. Also given that only a few people are actually adding data a lot of people need to interpret these models without actively producing them.
- Waiting: It can be a challenge when a change is done for this to propagate also to e.g., ERP systems and supplier systems. For some type of data this propagation is automatic.
- Extra processing: Necessary to add data up front. Can be a challenge when you need to perform changes, to have the data produced in earlier phases updated.

5 Conclusion

Above, we have seen three assessments done using the specialization of SEQUAL for data quality of specifically relevant knowledge sources to be used in a Virtual Obeya. This has highlighted opportunities, but also challenges when trying to integrated data from different knowledge sources typically used by people in different roles in an organization in a common user interface, supporting collaboration. In particular it highlights how different tools have a varying degree of explicit meta-model (data model), and that this is available in a varying degree. E.g., in many export-formats one loses some of the important information on product data. Even when different tools support e.g., process data, it is often process data on different granularity. The tools alone all have challenges relative to waste in lean engineering. In a Virtual Obeya environment one would explicitly want to combine data from different sources in a context-driven manner to address these reasons for waste. Depending on the concrete data sources to combine, this indicates that it is often a partly manual job to prepare for such matching. Also the different level of agreement of data from different sources (social quality) can influence the use of schema and object matching techniques in practice.

As with the quality of a BPM [14] and data models [15], we see some benefit both for SEQUAL and for a framework for data quality by performing this kind of exercise:

- Existing work on data and information quality, as summarized in [2, 24, 25] can be positioned within the generic SEQUAL framework as described in Section 3.
- These existing overviews are weak on explicitly addressing areas such as empirical and social quality, as also described in Section 3.2 and Section 3.6. Guidelines and means for empirical quality can build upon work in data and information visualization.

- The work by Batini and Price et al. On the other hand enriches the areas of in particular semantic and pragmatic data quality, as described in section 3.4 and section 3.5.
- The framework, especially the differentiation between the different quality levels has been found useful in the case from which we have reported in Section 4, since it highlights potential challenges of matching data from different sources as discussed above. On the other hand, to be useful, an additional level of specialization of the quality framework was needed.

Future work will be to device more concrete guidelines and metrics and evaluate the adaptation and use of these empirically in other cases, especially how to perform trade-offs between the different data quality types. Some generic guidelines for this exist in SEQUAL [13], which might be specialised for data quality and quality of conceptual data models. We will also look at newer work [3, 9] in the area in addition to those we have mapped so far. Due to the rapid changes to data compared to conceptual models guidelines for achieving and keeping model quality might need to be further adapted to be useful when achieving and keeping data quality. We will also look more upon the use of the framework when integrating data from less technical areas such as CRM and ERP data.

Acknowledgements. The research leading to these results was done in the LinkedDesign project that has received funding from the European Union Seventh Framework Programme ([FP7/2007-2013]) under grant agreement n°284613

References

1. Aasland, K., Blankenburg, D.: An analysis of the uses and properties of the Obeya. In: Proceedings of the 18th International ICE-Conference, Munich (2012)
2. Batini, C., Scannapieco, M.: Data Quality: Concepts, Methodologies and Techniques. Springer (2006)
3. Batini, C., Cappiello, C., Francalanci, C., Maurino, A.: Methodologies for data quality assessment and improvement. ACM Comput. Surv. 41(3) (2009)
4. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language: User Guide, 2nd edn. Addison-Wesley (2005)
5. Francalanci, C., Pernici, B.: View integration: A survey of current developments. Technical Report 93-053, Politecnico de Milano, Milan, Italy (1993)
6. Grudin, J.: Groupware and social dynamics: eight challenges for developers. Communications of the ACM 37(1), 92–105 (1994)
7. Hermans, F.F.J.: Analyzing and Visualizing Spreadsheets. PhD thesis, Software Engineering Research Group, Delft University of Technology, The Netherlands (2012)
8. Høydalsvik, G.M., Sindre, G.: On the purpose of object-oriented analysis. In: Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA 1993), pp. 240–255. ACM Press (1993)
9. Jiang, L., Barone, D., Borgida, A., Mylopoulos, J.: Measuring and Comparing Effectiveness of Data Quality Techniques. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 171–185. Springer, Heidelberg (2009)

10. Krogstie, J.: Using Quality Function Deployment in Software Requirements Specification. Paper presented at the Fifth International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ 1999), Heidelberg, Germany, June 14-15 (1999)
11. Krogstie, J.: Evaluating UML Using a Generic Quality Framework. In: Favre, L. (ed.) UML and the Unified Process, pp. 1–22. IRM Press (2003)
12. Krogstie, J.: Integrated Goal, Data and Process Modeling: From TEMPORA to Model-Generated Work-Places. In: Johannesson, P., Søderstrøm, E. (eds.) Information Systems Engineering From Data Analysis to Process Networks, pp. 43–65. IGI (2008)
13. Krogstie, J.: Model-based development and evolution of information systems: A quality approach. Springer, London (2012)
14. Krogstie, J.: Quality of Business Process Models. In: Sandkuhl, K., Seigerroth, U., Stirna, J. (eds.) PoEM 2012. LNBIP, vol. 134, pp. 76–90. Springer, Heidelberg (2012)
15. Krogstie, J.: Quality of Conceptual Data Models. In: Proceedings 14th ICISO, Stockholm Sweden (2013)
16. Krogstie, J.: A Semiotic Framework for Data Quality. In: Nurcan, S., Proper, H.A., Soffer, P., Krogstie, J., Schmidt, R., Halpin, T., Bider, I. (eds.) BPMDS 2013 and EMMSAD 2013. LNBIP, vol. 147, pp. 395–410. Springer, Heidelberg (2013)
17. Krogstie, J., Arnesen, S.: Assessing Enterprise Modeling Languages using a Generic Quality Framework. In: Krogstie, J., Siau, K., Halpin, T. (eds.) Information Modeling Methods and Methodologies. Idea Group Publishing (2004)
18. La Rocca, G.: Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design. Advanced Engineering Informatics 26(2), 159–179 (2012)
19. Lillehagen, F., Krogstie, J.: Active Knowledge Modeling of Enterprises. Springer (2008)
20. Manyika, J., Sprague, K., Yee, L.: Using technology to improve workforce collaboration. What Matters. McKinsey Digital (October 2009)
21. Moody, D.L.: Metrics for Evaluating the Quality of Entity Relationship Models. In: Ling, T.-W., Ram, S., Li Lee, M. (eds.) ER 1998. LNCS, vol. 1507, pp. 211–225. Springer, Heidelberg (1998)
22. Moody, D.L.: Theorethical and practical issues in evaluating the quality of concep tual models: Current state and future directions. Data and Knowledge Engineering 55, 243–276 (2005)
23. Nelson, H.J., Poels, G., Genero, M., Piattini, M.: A conceptual modeling quality framework. Software Quality Journal (2011)
24. Price, R., Shanks, G.: A Semiotic Information Quality Framework. In: IFIP WG8.3 International Conference on Decision Support Systems (DSS 2004), Prato, Italy, July 1-3, pp. 658–672 (2004)
25. Price, R., Shanks, G.: A semiotic information quality framework: Development and comparative analysis. Journal of Information Technology 20(2), 88–102 (2005)
26. Recker, J., Rosemann, M., Krogstie, J.: Ontology- versus pattern-based evaluation of process modeling language: A comparison. Communications of the Association for Information Systems 20, 774–799 (2007)