H. Giese, K. Rosenthal (Hrsg.): Modellierung 2024 Satellite Events,
Digital Library, Gesellschaft für Informatik e.V., 2024 1

How to define the Quality of Data and Software Models? A Data Quality Perspective

Gabriele Taentzer[1], Arno Kesper[2], Markus Matoni[3]

Modeling has always been advocated as an important part of data and software development in order to manage complexity by providing abstractions and hiding technical details. Due to the wide application of modeling, numerous informal and formal approaches to modeling have been developed, such as Entity Relationship Diagrams for data modeling, the Unified Modeling Language for software development, and the Business Process Modeling Notation for business process modeling.

Models have been used for a variety of data and software engineering tasks, including the stakeholder communication, design thinking, example simulation, and test case and source code generation. To be effective in these tasks, models must be of high quality. So *what is model quality?* Reviews of the existing literature [Mo05, MDN09] shows that the quality of data and software models is a multifaceted concept with multiple dimensions. However, the definition of quality dimensions varies widely.

Moody & Shanks [MS94] were the first to present a catalog of quality dimensions for *data models*. This catalog is well accepted in the literature. It contains very general and broad definitions of data model quality. Moody & Shanks defined model quality in terms of a specific purpose of modeling (e.g., specifying user requirements) and distinguished the quality dimensions *simplicity, completeness, flexibility, integration,* and *understandability.* Simplicity refers to the complexity of a model; completeness refers to the ability to meet all user information and functional requirements; flexibility refers to the ability to adapt to change; integration refers to the consistency of the modeled data with the rest of the organization's data; and understandability refers to the ease with which the model can be understood by the users of the model.

Mohaghegi et al. [MDN09] present the results of a systematic review of the literature discussing model quality in *model-based software development*. Model quality is contextually defined for a generic "purpose of modeling" and with a generic "level of abstraction". A generic definition of the purpose is advantageous because, as mentioned above, models can serve very different purposes. The level of abstraction also depends on the context

[1] Philipps-Universität Marburg, Germany, taentzer@mathematik.uni-marburg.de
[2] Philipps-Universität Marburg, Germany, arno.kesper@uni-marburg.de
[3] GWDG, Germany, markus.matoni@gwdg.de
 https://doi.org/10.18420/modellierung2024-ws-010

of a model, which can be, for example, requirement specifications, system architecture design, business process design, and others. The following dimensions of model quality have been identified: *correctness, completeness, consistency, comprehensibility, confinement,* and *changeability.* Syntactic correctness is defined as not violating the language syntax, rules and guidelines; semantic correctness refers to modeling our understanding and the domain in the right way; completeness is defined as having all necessary information that is relevant; consistency refers to having no contradictions in the model; comprehensibility refers to being understandable by the intended users (which may also be tools); confinement is defined as being consistent with the purpose of modeling and the type of system; and changeability is defined as supporting changes or improvements so that models can be changed or evolved quickly and continuously. Because models can be used in so many different ways, all of these dimensions are defined according to the purpose of modeling and the level of abstraction. The *purpose of modeling* determines which aspects of the domain of interest are considered critical. The *level of abstraction* is used to select an appropriate level of detail. *A comparison of the quality definitions for models used in data and software engineering and in other domains is still open.*

Given that *data and software models are structured data*, it is also interesting to consider the relationship between model and data quality. As representative of both quality definitions [MS94, MDN09], we will relate Mohaghegi et al.'s definition of model quality to the ISO standard [IS11] for data quality. This standard distinguishes between two main types of quality dimensions: *intrinsic dimensions* refer to the quality of the data itself, while *contextual dimensions* refer to the data in a given context. Looking at the quality dimensions in [MDN09], *correctness, completeness,* and *consistency* refer directly to accuracy, completeness, and consistency, which are classified as intrinsic dimensions in ISO. *Confinement* refers to a form of conciseness and is most likely comparable to efficiency in ISO. Contextual dimensions were not the focus of Mohageghi et al. with the exception of *comprehensibility*. Instead, they mention the quality dimension *changeability* as the ability of the model to change rapidly and continuously. We believe that a high quality model is generally easier to change and therefore classify *changeability* as a second-order dimension. *Models, like data, should also have a comprehensive contextual quality to enable effective work with them, including dimensions such as* accessibility, currency, plausibility, *and* traceability. These quality dimensions were not considered in [MDN09]. A recent paper by Firmani et al. [FTT19] considers additional dimensions of data quality, namely ethical dimensions. Specifically, they defined the dimensions of fairness, transparency, diversity, and data protection. *These dimensions address ethical challenges for source selection and knowledge extraction, among others, and may need to be adapted for model quality.*

Because models can serve very different purposes, the definition of model quality must be *configurable* so that the models are "fit for purpose." This fitness includes choosing the right level of abstraction. Using the goal-question-metric approach [CR94], the general quality dimensions can be considered to define quality goals, which are refined into constraints that specify purpose-specific questions. Defining model quality is the first step in a quality

assurance process. We then need to specify quality assessment for models. Based on the results of the model assessment, appropriate improvement steps should be considered. For example, a tool environment for *model quality assurance* based on the Eclipse Modeling Framework (EMF) is presented in [AT13]. The model assurance process can be customized to meet domain-specific and even project-specific needs. It supports the reporting of model metrics, the detection of model smells with patterns, and the refactoring of EMF models. A very different example for ensuring model quality is the slicing of MATLAB Simulink models as presented in [RG12].

Many new scenarios and tools in model-driven software engineering (MDSE) have emerged that benefit in some way from the application of *machine learning*. For example, modeling bots that play the role of virtual modeling assistants, inferring a model from a set of unstructured data, and even reviewing models in real time. Most of these applications typically require that high-quality models are available as training data to obtain high-quality learning results. [Ca18] A promising work in this direction is *ModelSet* [LIC22], a labeled dataset of software models for the use of machine learning in MDSE. *It is still open to analyze such model sets from the perspective of model quality.*

Bibliography

[AT13] Arendt, Thorsten; Taentzer, Gabriele: A tool environment for quality assurance based on the Eclipse Modeling Framework. Autom. Softw. Eng., 20(2):141–184, 2013.

[Ca18] Cabot, Jordi; Clarisó, Robert; Brambilla, Marco; Gérard, Sébastien: Cognifying model-driven software engineering. In: Software Technologies: Applications and Foundations: STAF 2017 Collocated Workshops, Revised Selected Papers. Springer, pp. 154–160, 2018.

[CR94] Caldiera, Victor R Basili-Gianluigi; Rombach, H Dieter: Goal question metric paradigm. Encyclopedia of software engineering, 1(528-532):6, 1994.

[FTT19] Firmani, Donatella; Tanca, Letizia; Torlone, Riccardo: Ethical dimensions for data quality. J. Data and Information Quality, 12(1), December 2019.

[IS11] ISO: Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Data quality model. Standard, International Organization for Standardization, Geneva, CH, December 2011.

[LIC22] López, José Antonio Hernández; Izquierdo, Javier Luis Cánovas; Cuadrado, Jesús Sánchez: ModelSet: a dataset for machine learning in model-driven engineering. Software and Systems Modeling, pp. 1–20, 2022.

[MDN09] Mohagheghi, Parastoo; Dehlen, Vegard; Neple, Tor: Definitions and approaches to model quality in model-based software development - A review of literature. Inf. Softw. Technol., 51(12):1646–1669, 2009.

[Mo05] Moody, Daniel L: Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. Data & Knowledge Engineering, 55(3):243–276, 2005.

[MS94] Moody, Daniel L.; Shanks, Graeme G.: What Makes a Good Data Model? Evaluating the Quality of Entity Relationship Models. In: Entity-Relationship Approach - ER'94, Business Modelling and Re-Engineering. LNCS 881. Springer, pp. 94–111, 1994.

[RG12] Reicherdt, Robert; Glesner, Sabine: Slicing MATLAB simulink models. In: 2012 34th International Conference on Software Engineering (ICSE). IEEE, pp. 551–561, 2012.