Complex Systems Informatics and Modeling Quarterly (CSIMQ)
eISSN: 2255-9922
Published online by RTU Press, https://csimq-journals.rtu.lv
Article 173, Issue 31, June/July 2022, Pages 1–17
https://doi.org/10.7250/csimq.2022-31.01

Metrics to Estimate Model Comprehension Quality: Insights from a Systematic Literature Review

Jordan Hermann[1], Bastian Tenbergen[1*], and Marian Daun[2]

[1]State University of New York at Oswego, USA
[2]University of Duisburg-Essen, Germany

hermann@oswego.edu, bastian.tenbergen@oswego.edu,
marian.daun@paluno.uni-due.de

Abstract. Conceptual models are an effective and unparalleled means to communicate complicated information with a broad variety of stakeholders in a short period of time. However, in practice, conceptual models often vary in clarity, employed features, communicated content, and overall quality. This potentially impacts model comprehension to a point where models are factually useless. To counter this, guidelines to create "good" conceptual models have been suggested. However, these guidelines are often abstract, hard to operationalize in different modeling languages, partly overlap, or even contradict one another. In addition, no comparative study of proposed guidelines exists so far. This issue is exacerbated as no established metrics to measure or estimate model comprehension for a given conceptual model exist. In this article, we present the results of a literature survey investigating 109 publications in the field and discuss metrics to measure model comprehension, their quantification, and their empirical substantiation. Results show that albeit several concrete quantifiable metrics and guidelines have been proposed, concrete evaluative recommendations are largely missing. Moreover, some suggested guidelines are contradictory, and few metrics exist that allow instantiating common frameworks for model quality in a specific way.
Keywords: Model-Based Development, Model-Based Engineering, Model-Based Software Engineering, Graphical Representations, Model Comprehension, Model Quality, Literature Survey.

1 Introduction

Model-based software engineering has established as avenue to remedy challenges in the development of modern software systems [1]. This includes the increasing demand for tightly integrated development processes taking different professions into account [2]–[5], the rapid need for innovation [6], growing inter-connectedness of devices [7], or the rise of cloud computing [8].

* Corresponding author

Reference: J. Hermann, B. Tenbergen, and M. Daun, "Metrics to Estimate Model Comprehension Quality – Insights from a Systematic Literature Review," Complex Systems Informatics and Modeling Quarterly, CSIMQ, no. 31, pp. 1–17, 2022. Available: https://doi.org/10.7250/csimq.2022-31.01

Additional information. Author ORCID iD: B. Tenbergen – https://orcid.org/0000-0002-0145-4800, M. Daun – https://orcid.org/0000-0002-9156-9731. PII S225599222200173X. Received: 9 June 2022. Revised: 19 July 2022. Accepted: 19 July 2022. Available online: 29 July 2022.

However, the term "model-based" can mean different things, depending on the specific sub-discipline of software engineering: while traditionally, the term was used to denote mathematical models of system execution (see, e.g., [9]), especially since the advent of UML, visual models and graphical representations are seen as a means to improve communication among stakeholders about the system under development (SUD) [1], [10], [11]. The particular benefit of conceptual models lies in their ability to structure complex information and dependencies through abstraction and through features of the modeling language. In the remainder of this article, we use the terms "conceptual model" and "visual diagram" interchangeably to mean graphical representations used to facilitate the development of complex software systems. The models under investigation in this work hence comprise any diagram types that may foster visual inspection by stakeholders, such as ArchiMate or UML, even though their primary purpose may be that of automated verification or execution (as is the case with graphs or SIMULINK models).

The benefit of conceptual models during development can only be harvested if the adequate modeling languages are used and if the created conceptual models are comprehensible. While research regularly deals with the former (i.e., when to use which modeling language in specific development situations), the question of when a diagram (of a given language) is comprehensible and therefore most useful to stakeholders during development is a question not easily answered.

Several quality frameworks have been introduced to assess the quality of diagrams for a specific purpose (e.g., [12]), to assess the usability of the notation (e.g., Moody's "Physics of Notation" [13]), or to assess the suitability of the diagram with regard to the universe of discourse (e.g., the FRISCO Report [14]). However, these frameworks are generic and difficult to operationalize regarding a specific development project. In consequence, modelers are faced with the challenge of deriving their own metrics to measure the quality of a conceptual model, which is often implicit to the modeler. To this day, there is still little guidance to assess whether a conceptual model is comprehensible. Nevertheless, several studies into factors impacting diagram comprehension have been conducted over the past decades. Still a wholistic summary on factors influencing model comprehension and metrics to measure model quality is missing.

In the following, we will focus on the comprehension quality of the model, not on other factors influencing the comprehension such that are inherent to the model reader, e.g., time constraints or experience level. To structure the body of work in this area and showcase guidelines for creating comprehensible conceptual models, this article contributes a systematic literature review to identify and to investigate metrics to measure comprehension quality of conceptual models.

This article is structured as follows: Section 2 introduces related work on model quality frameworks and factors influencing model comprehension. Section 3 presents the study design and Section 4 the major results of the literature review. Section 5 discusses implications before Section 6 concludes the article.

2 Related Work

Several prominent frameworks to assess the – broadly speaking – quality of models have been suggested. In a large body of work, Wand and Weber [16]–[24] build a model quality framework based on Bunge's Ontology [24], commonly referred to as the Bunge-Wand-Weber framework [25]. This framework differentiates the compositional view onto the SUD from the view of what the system externally interacts with. This differentiation allows assessing "internal" and "external forces" that affect the model of the system, hence reducing the question of "model quality" mostly to a question of "model completeness" and by extension "model correctness". This requires the reader to have a complete understanding of the domain for which the model was built.

By contrast, the FRISCO report [14] introduces a classification scheme for the basic elements of every model, covering semantic and ontological aspects of model creation. While more focused on the properties of the model itself, the framework must be instantiated for each model type specifically, requiring the quality assessment to become a question of adequacy of used modeling concepts. Recognizing this limitation, an extension of the FRISCO report is suggested by Krogstie

et al. [12], which propose five additional facets of model quality in addition to the semantic quality. These are: empirical (is the model suitable?), syntactic (are the model elements used correctly?), pragmatic (does the model say what needs to be said?) and social (do the modeler and the reader understand it the same way?). The authors point out the difficulty in creating an absolute measure of quality, meaning that their framework also needs to be instantiated for each model separately.

A more generically applicable framework is Moody's "Physics of Notation" [13]. In this work, Moody proposes several "principles" to assess the adequacy of the used modeling concepts to represent the subject matter (e.g., "Cognitive Fit" to describe if the used symbol is adequate to communicate a concept). These principles, however, serve mainly as guidelines to improve the visual aspects of newly created modeling language as opposed to assessing the quality of a model to represent the requirements of a system under development. Moody's work was extended in the framework by Genero, Piattini, and Calero [26], which aims to combine Moody's focus on the development of visual models with measures to improve layout and positioning of elements within the models. Combining both methods was meant to create a complete measure of diagram comprehensibility. The measures focus on four key areas of improvement. These are associations between model elements, aggregation of similar concepts, generalization to more abstract or reversely, to more specific concepts, and dependencies between concepts such as mutual constraints.

All these frameworks are similar in their argument about the individual model and how it is made, not what the role of the model is. This shortcoming is addressed in Kruchten's 4+1 View Model of Architecture [27]. To assist a selection of the model type and constrain the content this model ought to have, given the vastness of the subject matter, Kruchten structures his framework according to software development processes they support. Hence, the framework differentiates between the logical view (relationships between entities of the subject matter), the process view (functions and processes the SUD offers or is part of), the physical view (describing how the SUD fits into its operational context), the development view (the relationship between the system and the context in which it is developed), and the usage view (the application of the system). Although Kruchten's work was a key factor enabling model-driven software development by suggesting more concrete roles for diagrams, his framework limits the assessment of model quality to that of applicability if the underlying development process is roughly compatible.

In addition to the classic frameworks suggested above, a number of studies have taken a more empirical approach to find properties of models that indicate quality. For instance, MacCreery and Tenbergen [28] investigate students' use of various diagram types in different modeling contexts (e.g., capstone projects or assignment sheets) across several semesters and different courses. Using the framework by Krogstie et al. as the evaluative standard, the authors identified that in free modeling tasks, students rarely make use of all diagram features and, especially in capstone-type contexts, resort to using few model types extensively. The authors also identify common errors, such as missing association labels or confusing dynamic and static UML model elements.

El-Attar studied the model creation and comprehension process of roughly 150 students and practitioners [29], [30]. Participants were exposed paper-based or screen-based depictions of state charts and use case diagrams and were given comparatively simple comprehension tasks. These partly used models containing slight syntactical errors. Results show that students perform comparably to practitioners in model comprehension [29]. Furthermore, results show [30] the screen-based exposure to models may improve comprehension time and induce fewer comprehension errors than reading models on paper. Similarly, Zayan et al. [31] showed in an experiment involving 26 graduate students that participants were up to 80% less likely to make mistakes during model comprehension when provided with illustrative examples and asked up to 90% fewer clarification questions to domain experts.

In a recent and comprehensive study by Daun et al. [32], the authors collated the results of a series of experiments investigating the role of experience and confidence on model comprehension. In four experiments involving between 55 and 238 undergraduate and graduate students, participants were asked to self-assess their experience and confidence regarding visual

notations, exposed to functional design diagrams and message sequence charts, and asked a series of simple questions pertaining to the content of the diagrams. Results show that while most students have an implicit "feeling" of correct versus incorrect diagrams, their self-reported experience and confidence do not adequately predict performance. Yet, these factors are often used as a measure to establish participant comparability in empirical studies.

In summary, very little work is apparent that directly aims to operationalize the several frameworks that exist to quantify model quality. In addition, empirical work largely appears to investigate properties of the modeler or the modeling process, rather than investigate the models as the product. An exception is the work [28], which, however, is also agnostic of the aforementioned quality frameworks. In conclusion, there is little knowledge available on how the quality of the diagram itself impacts comprehension, or which properties a model should have in order to maximize understanding.

3 Study Design

To close the gap in research outlined above, we conducted a systematic literature review with the aim to provide an analysis of the state of the art on factors influencing model comprehension. In this section, we elaborate on the study design. The following Section 4 discusses the results.

3.1 Goal and Research Questions

In this study, we investigate metrics to measure comprehension of conceptual models. By this we mean any empirical finding that can be applied to a model to improve comprehension. This definition expands the range of applicable findings and allows for the inclusion of many smaller findings that may not make it into a normal strictly quantifiable metric. Locating all the findings with significance will indicate areas of modeling, during the modeling process, that may serve to improve comprehension. We define three research questions (RQs):

RQ1: What model types and model properties are measured in literature? Since the quality of models depends on the information expressed therein, this RQ aims to establish which types of models or modeling languages are subject to investigation.

RQ2: What metrics for model comprehension exist? Model types and modeling languages pertain to different conceptual levels and lend themselves to express different information but are united in their aim to convey information about the SUD. This RQ seeks to understand which concrete metrics can be proposed to quantify or guide the factors maximizing the readers' ability to comprehend the conveyed information.

RQ3: How are metrics quantified and which evaluative recommendations exist for them? This RQ investigates if the literature suggests any evaluative recommendation that may help in assessing the quality of models, either absolutely (e.g., no more than 7 classes per diagram) or relatively (e.g., fewer associations are preferable) given the factors found in RQ2. To understand this, we also take a closer look at the mode of quantification for the individual metrics.

3.2 Search Process

The study was planned as database search based on the best practices outlined by Kitchenham [33] and Petersen et al. [34], and – due to the thematic similarity – Genero et al. [36]. Only peer-reviewed digital sources, including journals and conferences, with online databases were searched. The electronic selections were made based on their inclusion of a broad range of topics in computer science. The search string (see Table 1) was created based on the primary search terms, these terms being "Comprehension", "Quality", and "Model/Diagram". These search strings were made using Boolean AND, OR terms to combine major search terms. For each term, synonyms have been defined, the search string has then been evaluated and optimized using the quasi-gold standard [35]. The search was conducted in commonly recommended databases such as ACM, IEEE,

SpringerLink, ScienceDirect, GoogleScholar. With databases that did not allow Boolean search terms the search strings were modified and the search system was manipulated to receive comparable results.

Table 1. Details on the search strings used. These were used in combination, as a single large string.

Term	Synonyms in Search String
Comprehension	Perceptions OR recall OR recall task OR semiotic OR survey OR understandability
Quality	Usability
Model/Diagram	UML OR BPMN OR FSA OR FST OR mealy machine OR moore machine OR class OR activity OR sequence OR i* OR i-star OR KAOS OR tropos OR GSN OR goal structure notation OR automat OR representation OR entity relationship OR EER

3.3 Inclusion / Exclusion Criteria

For inclusion decision was made to focus mainly on papers proposing or applying metrics to measure model quality, specifically those pertaining to comprehensibility. Papers not falling into this category were discarded after discussion between the first two authors. Papers addressing factors pertaining to the modeler (e.g., level of experience) were excluded in the same manner. We excluded paper from the result set if they did not consider graphical models (e.g., Matlab models of system execution) or did not focus on visual aspects of the diagram fostering human comprehension (e.g., suitability of class diagrams to be transformed into Entity Relationship diagrams). Furthermore, we excluded all non-peer reviewed work, contributions to fields other than computer science and software engineering, papers neither available online nor via interlending, non-English publications, papers fewer than 4 pages, editorials, opinion pieces about modeling practices without discussion of concrete metrics, and papers from before 1995 as we are interested in metrics going beyond the established model quality frameworks. The search took place in summer and fall 2018 and includes studies until that time.

3.4 Paper Selection Process

Three stages of filtering were conducted to reduce the set of papers: first, papers were selected based on the title; second, based on the abstract; third, based on the actual content. For each database, the results were sorted according to relevancy and the first 150 results from each database that we searched were selected for review by title search. This process involved one researcher sorting article relevance by title and another researcher reviewing this sort and comparing the decisions made; any conflicts were discussed and if no decision could be found the paper was included. From title filtering 566 papers were identified as potentially relevant. Abstract screening resulted in excluding of 249 papers, leading to 317 remaining as potentially relevant. These were then investigated in-depth by reading the entire paper and identifying the proposed metrics within. In doing so, a set of 115 relevant papers that propose some kind of metric to measure model comprehension has been received, from which another 6 papers were excluded as duplicates or non-English papers. The final set of 109 papers was included in this study.

3.5 Validity Evaluation

The scope and size of this research leaves it open to several threats to validity. The discussion of the potential threats according to [36] and the measures to reduce or eliminate them follows.

Conclusion Validity. Regarding the conclusions that can be drawn, it must be admitted that throughout the developed guidelines for search and selection, there is a potential for bias to affect

the search. The use of two reviews and a policy of "include when in doubt" was adopted to reduce this threat. Since no classification of results beyond include/exclude took place, no interrater reliability measure was applied.

Internal Validity. The maturation of the researchers during the data extraction period makes it possible that papers reviewed later may be more thoroughly or effectively investigated than earlier searches. This means some results may not have been found properly. There was a review of all articles to attempt to mitigate this issue. The selection of articles and general searches may have been affected by internal bias. This is difficult to totally avoid, but the use of two reviewers limited the scope of this threat. The third researcher acted as quality control for this purpose.

Construct Validity. There is a potential for inadequate preoperational explication of constructs. During the creation of the search strings and research questions there is a chance that some key elements or concepts were overlooked. These questions were reviewed several times by all authors to reduce the chance of this occurring. One of the researchers' main fields of investigation is in UML and its application of it in the development process. This may have introduced a bias into the creation of RQs. These research questions were defined regardless of application of the model and focused on comprehension to avoid the mono-operations bias. Repeated reviews of activities by all researchers reduce, but do not remove this threat. The entire survey is vulnerable to restricted generalizability across constructs. The large number papers collected, and the cross comparisons mitigates this issue. The non-specific nature of the heuristics presented also reduces potential for inadequate explication of constructs.

External Validity. The lengthy review process mixed with the general learning curve for this type of research opens the study up to the potential of history effect findings. The articles were reviewed in order and the re-reviewed after completion of the first cycle, to reduce the chance of experience effecting results.

4 Results

4.1 RQ1: Model Types and Model Properties

To answer RQ1, we first quantified the number of times primary studies investigated specific model types (e.g., BPMN or UML use case diagrams) and which model properties these studies investigated (e.g., compositionality of items or use of stereotypes). For each, we also identified what type of evidence the study at hand proposes (e.g., correctness of usage or experience report). In all cases, we resorted to the study's explicit mention of model types, properties, and characterization of evidence, allowing for multiple matching categories (e.g., if a study compared the effect of layout on class diagram and ER diagram comprehension), but combined several papers by the same authors to a single study if reporting on the same or similar topic (e.g., as is the case with extended versions of manuscripts). A comparatively small number of studies present no evidence beyond author opinion, noted under category "opinion" [34]. Figure 1 shows the results.

As Figure 1 shows, one of the most researched areas in the field is on the comprehension of UML class diagrams. This focus is understandable given the popularity and wide use of these. The visual features of conceptual models also mainly focused on models of UML. Most other model types outside of UML and BPMN are ignored by research in general (e.g., KAOS diagrams) or, as is the case with EPC, YAWL, or CSD models, receive a comparatively little attention. Given that some model types such as KAOS goal diagrams, feature models, or GSN safety arguments are used for rather special purposes as opposed to generically applicable ER, UML, or BPMN models, this may explain the relatively few studies regarding their comprehensibility.

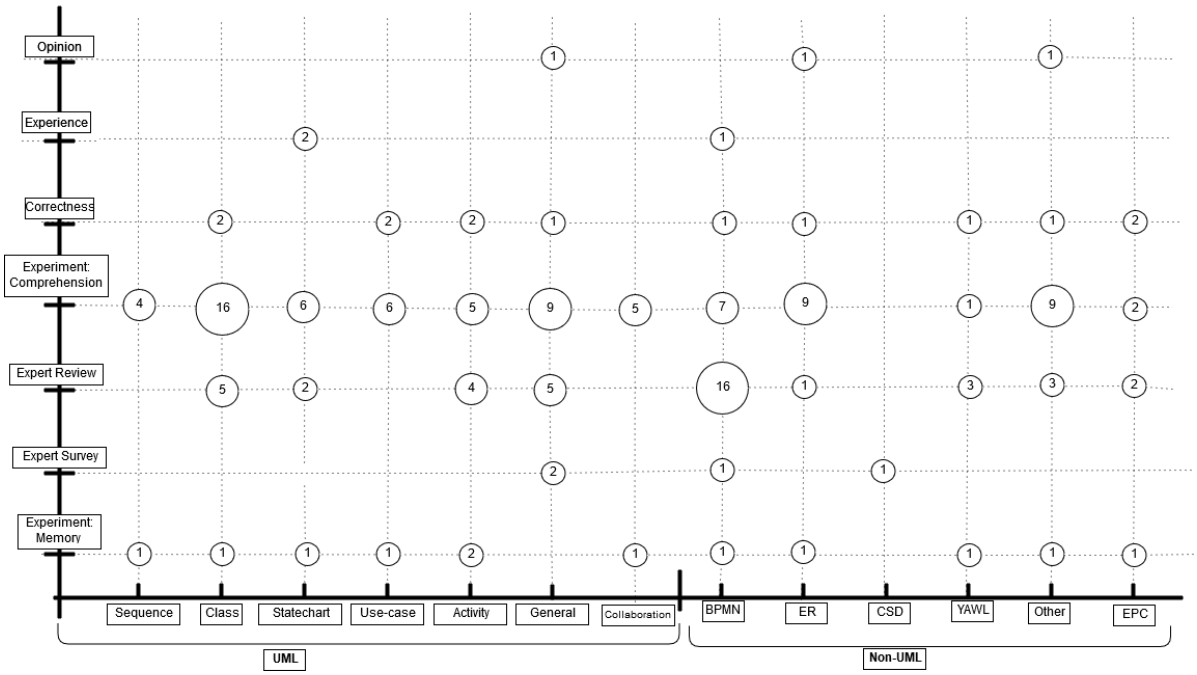

Figure 1. The number of papers found for each model type (divided between UML and non-UML languages)

Most studies aim to generically investigate "visual features" of models. In this category, we counted studies, which investigate aspects of the visual notational elements, such as shape of model elements, visual style of associations (e.g., bend, cornered, or Bezier-curved), or other visual features that are not strictly related to layout. Yet, in addition to layout in general as well as appropriateness of used symbols, visual features of diagrams are among the most intensively studied aspects of model comprehension, especially using empirical investigations. These three model properties are also the key drivers of Moody's Principle of semiotic clarity described in [12], [13]. Surprisingly many studies specifically investigate the use of stereotypes in UML diagrams or composite states in automata model types, while other specific model properties (e.g., n-ary relationships in class and ER diagrams) have not been the subject of identified primary studies (composite states in this case is not to be confused with nestedness of model elements, which partly includes containment- and aggregation relationships).

Answer to RQ1: From these findings we can conclude that the predominant factors pertaining to model comprehension that are investigated in the literature are visual features as well as layout of UML class diagrams and BPMNs, followed by remaining UML diagram types.

4.2 RQ2: Metrics for Model Comprehension

To answer RQ2, we quantified the number of times primary studies investigated specific model properties (e.g., stereotypes in UML models or visual features such as cornered associations between symbols vs. Bezier-curved associations). We quantified these studies using the type of evidence identified in RQ1 (see Figure 1). Results are shown in Figure 2. As can be seen, the majority of studies identified in Figure 2 experimentally investigate the comprehension of visual features, layout, modeling symbols, and UML stereotypes in order to make concrete proposals for easy-to-understand models. By comparison, experimentally investigating the impact of these properties on memory or any other stakeholder factor is seldom studied. This plethora of empirical evidence is supplemented with less rigorous and also less plentiful evidence based on author opinions or experiences or evidence gathered by querying experts.

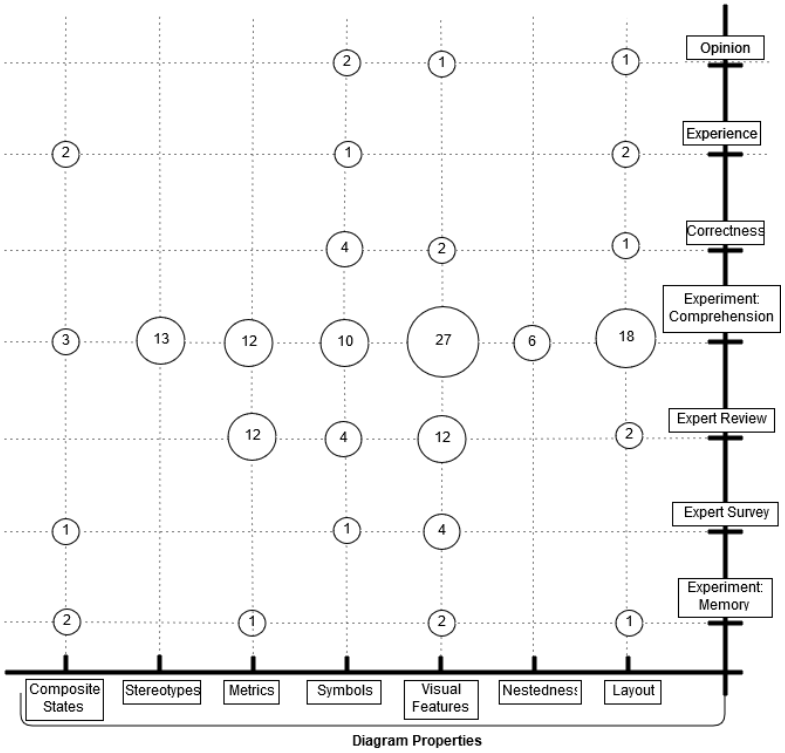

Figure 2. Model properties concerned by specific research methods

Thus, the evidence on factors influencing model comprehension (i.e., not factors pertaining to the modeler or model reader, such as "experience with models") is largely empirically substantiated and often aims to quantify the amount of certain modeling elements or define naming conventions for them. An overview over all used factors is given in Table 2. Under "source" examples of studies are given which suggest a concrete metric that match the description. Table 2 also highlights concrete ways to quantify model properties or identifies guidelines which impact model comprehension and relates these to similar metrics given a quality property the metrics pertain to. In the following, we will discuss findings regarding the impact of these metrics in more detail, highlighting common themes in the quality metrics.

A series of identified primary studies focus on ways to quantify ***model size*** and ***complexity***. In [49] and [52] emphasis is on the general relationships within a model. Note that while size and complexity are closely related concepts, they are not the same: *size* refers to the number of used modeling elements while *complexity* refers to the number of connections between them. This broad focus holds all types of relationships as equally contributing to model complexity. This method makes for faster measuring, but it is not clear if all relationship types add equally to model complexity. Works such as [50], [51], and [53] measure each individual type of relationship separately, giving different weights to different types of relationships. This method allows for a more complete and descriptive measurement on cost of complexity of the measurement itself. [62] attempts to measure both total relationships and the number of each type. However, while it is clear the number of relationships in a diagram has an impact on comprehension; it is less clear if there are specific effects for each type of relationship.

Metrics measuring ***naming conventions*** directly attempt to increase comprehension by setting out a clear language for a model. There are two primary ideas around this. Firstly (see, e.g., [57]), focusing on the development of a set of terms and shorthand clues within a group of modelers and readers allows establishing quality guidelines to ease comprehension and communication. However, doing so means that model quality is specific to a set of readers and their shared application domain, perhaps at the expense of involving external readers or onboarding new team members. Through various means a consensus is drawn about the meaning and then applied to all models in the company or group. Furthermore, (see, e.g., [38]) models can be constrained in a way

similar to how constrained natural language results in less ambiguity in natural-language requirements. Doing so leverages ease of comprehension by anyone knowing the modeling language, however, reduces the expressive power of the models.

Table 2. Metrics for model comprehension, quality property (QP) they pertain to, and paper source

QP	Metric Description, Quantification, or Guideline	Source (example)
Model Size	Number of specific relationships to measure complexity of a model, e.g., 1:1 or 1-n relationships.	[50], [54]
	Number of classes and their responsibilities.	[51], [52]]
	Number of aggregations and hierarchies. / Number of used associations.	[49]
	Number of relationships / Zhou's metric measures relationships but then uses a formula to transform the diagram into a weighted graph of relationships to improve comprehension.	[62], [63], [81]
	Size of the model in terms of number of classes, relationships, and aggregations.	[62]
Naming	Organization-specific guidelines for labeling elements to improve communications and understanding.	[56]
	A set of three different methods of applying style choices, each with a slightly different method of explaining and naming elements.	[38]
Semantics	Guidelines for establishing correct and effective terms and language in a model. A set of universal terms that are modified on a company-by-company basis.	[66]
	Guidelines to create a semiotically clear language and model.	[79]
	The degree to which symbols can be recalled.	[48]
	A set of in-company guidelines of effective construction of elements and symbols to be used when creating diagrams.	[56]
	Rufai's metric to compare the similarity between two models	[81]
Model Complexity	In's metric to measure the complexity using relationships by counting the number of roles, parameters, and operations.	[81]
	Number of entry and exit actions in a diagram as an element of complexity that negatively affects comprehension.	[53]
	Amount of information provided to describe or explain the elements in a diagram.	[65]
	The degree to which external resources are needed to explain a model.	[72]
	The degree of nestedness in a diagram.	[46]
Layout	Proximity of elements in a model.	[38]
	Number of nodes and edges. / Proximity of elements. / Number of line bends.	[41]
	Number of crossing edges.	[55]
	Using Gestalt principles to develop more clear and comprehensible models.	[60],[73]
	Proximity of elements. / Use and size of element clusters. / Degree of nestedness of elements.	[76], [77], [78], [80]
Use of Edges and Nodes	Degree of connectedness of model elements.	[39]
	Number of line bends and crosses. / Degree of uniformity of variables.	[40]
	Number of edges and nodes. / Names of edges.	[47]
	Placement of nodes and edges. / Proximity to nodes and edges.	[64]
	Number of forks, joins, and other aspects of edge use.	[70]
	Number of gateways, entries, and exits.	[74]
Stereo-types	Number of stereotypes.	[59], [68], [71], [75]
	Degree of cognitive clarity of stereotypes.	[76], [77], [78], [80]

Semantic metrics go beyond model feature restrictions. The aim of these metrics is to create both a system of labels and symbols, as well as a unique definition of their meaning. This eliminates elements with the same meaning as well as elements with redundant concepts, Hence, having a similar effect on synonyms and homonyms like constrained naming conventions do. For instance, in [66], a wide set of symbols and labels is created that cover as many aspects of the universe of discourse as possible, creating a broad set of standards that can then be individually modified within a company as needed. Again, this can greatly improve intra-organizational communication but run the risk of causing confusion with inter-organizational communication.

The work in [79] focuses on streamlining BPMN diagrams for inter-organizational cooperation. This includes reducing the number of repeated elements and creating universal shorthand signs. In [56], an example of a semiotic language extension in use for a single company is given. This provides a good base for further development as it outlines common issues and gives actionable solutions. The authors in [48] investigated the recallability of symbols. Setting out clear measure of recallability is essential to creating semantically effective languages and signs. The work in [81] defines a metric for comparing two models. This aims at reducing overlap and repeated information in complex, multi-model projects. By comparing the models, developers can understand how much overlap had occurred and improve the model to more specifically explain the universe of discourse yet may introduce modeling overhead due to the effort of creating multiple models on the same subject matter.

Nestedness of elements seeks to reduce ***model complexity*** by grouping elements into smaller, related sections. For instance, [42] and [43] investigated the effect of nestedness in different groups of modelers. A critical area of understanding is how effective a given method is at improving comprehension. Their results show that nestedness is a key tool for developing effective models. The authors in [46] set out to create a set of measurements that can be used to gauge the nestedness of a diagram. It also includes metrics to measure the effect of nestedness on comprehension. In [44], a critical restriction on nestedness is added. The authors provide a clear use for the previous metric. Their results indicate that only a small level of nestedness is effective. Increasing levels may decrease comprehension and lead to negative effects.

Papers concerning the ***layout*** seek to optimize the physical location of elements. In [38], a set of guidelines on how to place elements when creating a diagram is proposed. The authors compare thresholds of too tight or too loose grouping of elements and identify that both may impair a reader's ability to see causal connections between elements. [41] propose a method to quantify elements for this purpose, focusing on nodes, edges, and proximity of elements. The work in [55] and [57] looks at edge crossing and arrows and the effect on the overall comprehension. In contrast to approaches that take element proximity into account, these works assert that minimizing crossing lines improves comprehension. In [60] and [73], yet a different stance is taken. These sources are the only primary studies identified in our search that use Gestalt principles to improve comprehension.

Use of edges and nodes pertains to factors influencing model comprehension in ways beyond the additional complexity that increased use of edges between nodes usually entails. In this sense, [39] proposes a set of guidelines aimed at limiting and planning the placement of edges. These guidelines measure the proximity of each edge, the overall number, and the number of crosses, indicating that these should be limited as often as possible. [40] used eye tracking software to confirm the negative effects of line crossing on the ability of a reader to follow the flow of a diagram. [47] measures the total number of arcs and nodes and presents guidelines for labeling elements. The number of arcs is the same as the number of line bends in other works, such as in [64]. In this work, arcs as well as edge and node placement and proximity is quantified. Similarly, [69] counts the number of nodes, line bends, edges, and proximity to reduce all of these and improve comprehension. A unique approach is taken in [74], where quantification measures the number of splits, joins, and edges that can be reduced to improve comprehension. The authors also measure the number of gateways, entries, and exits in a diagram. Like other measures, the authors indicate that many these elements can decrease comprehension. Yet, therein a tradeoff exists between omitting these elements whenever possible while also maintaining the semantic precision regarding the subject matter. This is because leaving out, for instance, a diamond-shaped n-ary relationship in UML class diagrams and replacing it with, e.g., n binary relationships may increase semantic clarity of the relationships. However, this may be at the expense of model complexity and at the expense of perhaps no longer adequately reflecting the universe of discourse.

In this list of model quality properties, ***stereotypes*** are the only type of metric in Table 2 that pertains to specific model types since stereotypes are mostly used in UML diagrams. Yet, several studies have specifically indicated these as impactful means that influences model comprehension.

UML already makes some specific suggestions on the use of stereotypes (and in some cases requires it), but nevertheless, some studies provide empirical support for their adequate use. Examples include [59], [68], and [71]. These studies show that stereotypes can be an effective means to increase comprehension by providing further details on the nature of the entity they describe. It should be noted that [75] found this effect to be prevalent regardless of experience level of the reader of the model.

Answer to RQ2: Many concrete metrics mostly seeking to quantify model elements have been proposed, predominantly to assess size and complexity of the model, and investigate the quantification of diagram features such as stereotypes and edge features. Metrics pertaining to layout and model semantics resort to hard-to-quantify guidelines and subjective recommendations.

4.3 RQ3: Quantification and Evaluative Recommendations?

Some studies are rather explicit how to quantify their proposed metrics. An overview is given in Table 2. In the following, we discuss certain metrics that are similar in their quantification.

[52] breaks down its measurements into three categories, entity metrics, attribute metrics, and relationship metrics. Entity metrics measure the number of entities in an ER diagram. Attribute metrics measure the number of derived attributes, the number of composite attributes, the number of multivalued attributes, and the total number of attributes. Relationship metrics measure the total number of relationships, the number of m:n relationships, the number of 1:n relationships including 1:1, the total number of non-binary relationships, the number of binary relationships, the number of child-parent pairs across all generalization or specialization relationships, the number of reflexive relationships, the number of redundant relationships and a specific schema cohesion metric that measures the number of relationships that can be reached from a given relationship. [54] measures very similar aspects to [52], the number of entities, the number of attributes, the number of derived attributes, the number of composite attributes, the number of multi-valued attributes, the number of relationships, number of m:n relationships, the number of 1:n relationships, the number of n-ary relationships, the number of binary relationships, the number of reflexive relationships, the number of IS_A relationships. Both studies focus primary on relationships in ER diagrams to measure their size and, indirectly, complexity. The focus is on the same measurements in order to identify overly complex diagrams, asserting that smaller, less complex diagrams are easier to understand. A concrete evaluative recommendation is not given.

[51] measures the total number of classes, the number of packages in a system, the total number of root classes, the number of responsibilities, number of abstract responsibilities, number of immediate subclasses, and the number of dependencies. Many of these are combined into more abstract variables to quantify the model quality and make it comparable across the diagrams. Similarly, the work in [51] measures the number of associations, number of aggregations which is each whole-part pair, number of dependencies, number of generalizations looking at each child-parent pair, number of aggregations hierarchies, number of generalization hierarchies, the longest path from the root of a hierarchy to the class, the longest path from class to leaves, the total number of classes, the number of attributes, and the number of methods in the diagram. Again, [50] measures the total number of classes, the total number of attributes, the total number of methods, the total number of associations, the total number of aggregation relationships within a class diagram, the number of dependency relationships, the number of generalization relationships within a class diagram, the number of aggregation hierarchies, the number of generalization hierarchies, the longest path from root to class in the hierarchy, and the longest path from the class to the leaves. Both studies focus on more abstract connections within a diagram as a mode to quantify complexity, asserting that smaller models are preferable, yet without describing the impact on the model comprehension or a trade-off threshold when comprehension deteriorates.

The work in [56] takes a different approach and describes a set of naming and labeling conventions, rather than a mode of quantification. Twenty-three specific instructions are proposed. The goal of these instructions is to improve inter-organizational communication. By streamlining

the labels and names of diagram elements every viewer should be able to easily understand the diagram. As discussed before, there is no relative or absolute evaluative recommendation on which or how many guidelines should predominantly be used. [66] uses several kinds of guides to develop clear semiotic guidelines for communicating meaning creating new symbols. This framework comprises the goals of the organization, the set of all statements that can be made according to the graphemes, vocabulary, and syntax of the modeling language (which the framework calls "the language extension"), the domain of the model, the externalized models, the relevant explicit knowledge of the stakeholders, and the social actor interpretation. These metrics focus on the users understanding of the information presented. In these metrics, the viewers are the primary focus of understandability. Creating clear terms and symbols reduces the number of symbols used and any overlap. These metrics aim to correct the base language itself as opposed to the previous methods which focused on correcting each model as it was created.

As can be seen, only a small number of concrete relative or absolute guidelines or recommendations on which quantifiable model properties to optimize to foster comprehension are available. In consequence, there are few recommendations for the modeler to adhere to such that the reader is enabled to comprehend a model most easily. Nevertheless, some heuristics are proposed in the literature. For instance, [42] and [44] show that experienced users can take better advantage of composite states than inexperienced users. This claim is supported in [45]. The same authors also recommend that a low level of nesting is the most effective with higher levels affecting the comprehension negatively [43]. Heiser and Tversky [57] suggest that arrows as edge decoration can improve relationship comprehensibility in certain model types if the language allows this. Studies such as [76]–[78], and [80] argue that the use of clustering can improve the comprehension of class diagrams, even for nested modeling elements. This is particularly effective if stereotypes are used to clarify element composition [71], regardless of experience level of the model reader [75]. In an eye tracking study [80], these claims have been empirically supported. Larger models have a negative impact on comprehension. Moreover, [59] and [68] show that using stereotypes can increase the speed of comprehension as well as the accuracy, suggesting the more stereotype use is preferable.

There is a fine limit in scale. Too many elements make the model are too complex. With too few elements the diagram is too vague. While this may seem as an obvious heuristic, the work in [82] and [83] has substantiated this subjective impression and recommends finding a suitable diagram size depending on the universe of discourse, yet without giving an absolute or relative limiting threshold. Similarly, as Purchase et al. show [67], increased amounts of nodes, line bends and edges used in a model impact comprehension; as the number increases, comprehension decreases.

Answer to RQ3: Very few concrete quantifiable measures and evaluative recommendations exist that would help a modeler to optimize their models for comprehension. Naming conventions demonstrably improve inter-organizational comprehension, possibly, at the expense of onboarding cost of new team members. Model type specific heuristics exist and are (to some degree) empirically validated.

5 Discussion

For ***RQ1***, we investigated the field of research on model comprehensibility in terms of considered models used and model properties. We found out that, in general, the field has quite extensively been researched in the last decades as we found a relatively high number of relevant papers that define or investigate some kind of metric to assess the comprehension quality of models. However, research is highly concentrated into a few areas, such as UML class diagrams and BPMN diagrams, leaving large areas of model-based research unconsidered for model comprehension (e.g., graph representations). Regarding model properties, we found research mostly concerned with visual features and the impact of layout decisions.

For ***RQ2***, we investigated the proposed metrics and their empirical fundament. Studies use a wide variety of empirical approaches (e.g., structured expert analyses [39] or empirical experiments [38]) to investigate static aspects of specific diagram types. These include the number of line bends of connectors in ER diagrams [54], nesting level of states in UML state machine diagrams [44], naming of UML modeling elements [61], etc. However, only few of these studies provide tangible, operationalizable heuristics, which can be applied in practice (see [60] for a good example of heuristics to improve comprehension). Moreover, some findings seem to be contradictory. For instance, while [43] suggest avoiding nested states in behavioral diagrams, [78] and [58] report that nesting generally improves comprehension. In summary, we found most metrics to aim to quantify use of model elements to account for measuring size and complexity of models. However, also some metrics of model features that are typically hard to quantify have been proposed, e.g., regarding layout and clustering of elements. For these, basic guidelines and subjective recommendations are abundant, but almost exclusively require specific instantiation for different contexts and subject matters. Similarly, the use of stereotypes and naming conventions, appears, is most effectively improving comprehension within specific organizations.

For ***RQ3***, we investigated how the metrics concretely relate to model comprehension. We found only little substance on empirically proven sound recommendation systems for using the proposed metrics. For instance, while it is commonly accepted that the number of modeling elements in a model has an influence on comprehensibility as metric; it is often recommended to count the model elements or certain types of model elements. However, no rules exist, how many model elements pertain to a "good", that is "comprehensible", model.

6 Conclusion and Future Work

In this article, we contributed an analysis of approaches proposing and investigating model factors influencing model comprehension and metrics measuring comprehension quality of models by means of a systematic literature review. Our findings show that there exist a wide variety of such metrics. Our survey thus helps modelers in finding existing guidelines for their situations and provides researchers with a basic fundament to build upon further needed research on the effects of model comprehension. A limitation to our study is that the literature search concluded in 2018. Since then, empirical work on models and modeling processes have emerged, a selection of which was discussed in Section 2. Yet, a definitive set of guidelines for modelers remains to be desired.

It has also shown that further research is needed. Particularly, there is a need to combine existing metrics and to establish evaluative recommendation guidelines for using the metrics. There is very little guidance for the modelers on when a model is of good comprehension quality. Furthermore, the existing quality frameworks discussed in Section 2 are largely abstract and largely stand on their own. Albeit some of the same authors proposed further work on model quality and others make heavy reference to these frameworks, approaches, or guidance on how to instantiate the model quality frameworks by the modeler has yet to be established. Nevertheless, the existing model quality frameworks represent a good starting point. The immediate next step in this line of research is therefore to establish a knowledge base of existing metrics and the model types they are applicable to. In combination with the concrete metrics an instantiation of the frameworks can then be created to support modelers in assessing the comprehension quality of their model for a potential reader.

Acknowledgements

For a complete list of studies identified by our search, see the online supplement at: https://doi.org/10.6084/m9.figshare.19099745

References

[1] E. Sikora, B. Tenbergen, and K. Pohl, "Industry needs and research directions in requirements engineering for embedded systems," *Requir. Eng.*, vol. 17, no. 1, pp. 57–78, 2012. Available: https://doi.org/10.1007/s00766-011-0144-x

[2] B. Graaf, M. Lormans, and H. Toetenel, "Embedded software engineering: the state of the practice," *IEEE Softw.*, vol. 20, no. 6, pp. 61–69, 2003. Available: https://doi.org/10.1109/ms.2003.1241368

[3] C. McPhee and A. Eberlein, "Requirements engineering for time-to-market projects," *Ninth Annual IEEE Int. Conf. and Workshop on the Engineering of Computer-Based Systems*, 2002, pp. 17–24. Available: https://doi.org/10.1109/ecbs.2002.999818

[4] M. Weber and J. Weisbrod, "Requirements engineering in automotive development: experiences and challenges," *Proc. IEEE Joint International Conference on Requirements Engineering*, pp. 331–340, 2002. Available: https://doi.org/10.1109/icre.2002.1048546

[5] C. J. Neill and P. A. Laplante, "Requirements engineering: the state of the practice," *IEEE Softw.*, vol. 20, no. 6, pp. 40–45, 2003. Available: https://doi.org/10.1109/ms.2003.1241365

[6] H. Edison, N. bin Ali, and R. Torkar, "Towards innovation measurement in the software industry," *J. Syst. Softw.*, vol. 86, no. 5, pp. 1390–1407, 2013. Available: https://doi.org/10.1016/j.jss.2013.01.013

[7] M. Broy, M. V. Cengarle, and E. Geisberger, "Cyber-physical Systems: Imminent Challenges," in R. Calinescu, D. Garlan (eds) *Large-Scale Complex IT Systems: Development, Operation and Management*, Berlin, Heidelberg, pp. 1–28, 2012. Available: https://doi.org/10.1007/978-3-642-34059-8_1

[8] P. Mosterman and J. Zander, "Cyber-physical systems challenges: a needs analysis for collaborating embedded software systems," *Software & Systems Modeling*, vol. 15, pp. 5–16, 2016. Available: https://doi.org/10.1007/s10270-015-0469-x

[9] R. Bharadwaj and C. L. Heitmeyer, "Model Checking Complete Requirements Specifications Using Abstraction," *Autom. Softw. Eng.*, vol. 6, no. 1, pp. 37–68, 1999.

[10] I. Davies, P. Green, M. Rosemann, M. Indulska, and S. Gallo, "How do practitioners use conceptual modeling in practice?" *Data Knowl. Eng.*, vol. 58, no. 3, pp. 358–380, 2006. Available: https://doi.org/10.1016/j.datak.2005.07.007

[11] M. Lubars, C. Potts, and C. Richter, "A review of the state of the practice in requirements modeling," in *1993 Proc. of the IEEE Int. Symposium on Requirements Engineering*, pp. 2–14, 1993. Available: https://doi.org/10.1109/isre.1993.324842

[12] J. Krogstie, O. I. Lindland, and G. Sindre, "Towards a deeper understanding of quality in requirements engineering," in J. Iivari, K. Lyytinen, M. Rossi (eds) *Advanced Information Systems Engineering*, pp. 82–95, 1995. Available: https://doi.org/10.1007/978-3-642-36926-1_7

[13] D. Moody, "The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering," *IEEE Trans. Softw. Eng.*, vol. 35, no. 6, pp. 756–779, 2009. Available: https://doi.org/10.1109/tse.2009.67

[14] E. D. Falkenberg, *A framework of information system concepts: the FRISCO report*. Web edition, Leiden: University of Leiden, Department of Computer Science, 1998.

[15] Y. Wand, "Ontology as a foundation for meta-modelling and method engineering," *Inf. Softw. Technol.*, vol. 38, no. 4, pp. 281–287, 1996. Available: https://doi.org/10.1016/0950-5849(95)01052-1

[16] Y. Wand and R. Y. Wang, "Anchoring Data Quality Dimensions in Ontological Foundations," *Commun ACM*, vol. 39, no. 11, pp. 86–95, 1996. Available: https://doi.org/10.1145/240455.240479

[17] Y. Wand and R. Weber, "Mario Bunge's ontology as a formal foundation for information systems concepts," in *Studies on Mario Bunge's Treatise*, pp. 123–149, 1990.

[18] Y. Wand and R. Weber, "An ontological model of an information system," *IEEE Trans. Softw. Eng.*, vol. 16, no. 11, pp. 1282–1292, 1990. Available: https://doi.org/10.1109/32.60316

[19] Y. Wand and R. Weber, "Toward a Theory of the Deep Structure of Information Systems," *ICIS 1990 Proceedings*, p. 12, 1990.

[20] Y. Wand and R. Weber, "On the ontological expressiveness of information systems analysis and design grammars," *Inf. Syst. J.*, vol. 3, no. 4, pp. 217–237, 1993. Available: https://doi.org/10.1111/j.1365-2575.1993.tb00127.x

[21] Y. Wand and R. Weber, "On the deep structure of information systems," *Inf. Syst. J.*, vol. 5, no. 3, pp. 203–223, 1995. Available: https://doi.org/10.1111/j.1365-2575.1995.tb00108.x

[22] R. Weber, *Ontological Foundations of Information Systems*. Coopers & Lybrand and the Accounting Association of Australia and New Zealand, 1997.

[23] R. Weber and Y. Zhang, "An analytical evaluation of NIAM'S grammar for conceptual schema diagrams," *Inf. Syst. J.*, vol. 6, no. 2, pp. 147–170, 1996. Available: https://doi.org/10.1111/j.1365-2575.1996.tb00010.x

[24] M. Bunge, *Treatise on Basic Philosophy. Ontology I: The Furniture of the World*. Springer Netherlands, 1977. Available: https://doi.org/10.1007/978-94-010-9924-0

[25] A. L. Opdahl and B. Henderson-Sellers, "Ontological Evaluation of the UML Using the Bunge-Wand-Weber Model," *Softw. Syst. Model.*, vol. 1, no. 1, pp. 43–67, 2002. Available: https://doi.org/10.1007/s10270-002-0003-9

[26] M. Genero, M. Piattini, and C. Calero, *Metrics for Software Conceptual Models*. Imperial College Press, 2005. Available: https://doi.org/10.1142/p359

[27] P. Kruchten, "The 4+1 View Model of Architecture", *IEEE Software*, vol. 12, no. 6, pp. 42–50, 1995.

[28] J. MacCreery and B. Tenbergen, "On the Syntactic, Semantic, and Pragmatic Quality of Students' Conceptual Models," *Proc. 52nd Hawai'ian Intl. Conf. on System Sciences (HICSS-52)*, 2019. Available: https://doi.org/10.24251/hicss.2019.929

[29] M. El-Attar, "A Comparative Study of Students and Professionals in Syntactical Model Comprehension Experiments," *Software & Systems Modeling*, vol. 18, pp. 3238–3329, 2019. Available: https://doi.org/10.1007/s10270-019-00720-5

[30] M. El-Attar, "Are Models Better Read on Paper or on Screen? A Comparative Study," *Software & Systems Modeling*, 2022. Available: https://doi.org/10.1007/s10270-021-00966-y

[31] D. Zayan, A. Sarkar, M. Antklewicz, R. Maxiel, and K. Czarnecki, "Example-Driven Modeling: On Effects of Using Examplss on Structural Model Comprehension, What Makes them Useful, and how to Create Them," *Software & Systems Modeling*, vol. 18, no. 3, pp. 2213–2239, 2019. Available: https://doi.org/10.1007/s10270-017-0652-3

[32] M. Daun, J. Brings, P. A. Obe, and V. Stenkova, "Reliability of Self-Rated Experience and Confidence as Predictors of Students' Performance in Software Engineering," *Empirical Software Engineering*, vol. 26, no. 4, 2021. Available: https://doi.org/10.1007/s10664-021-09972-6

[33] B. Ktichenham, "Procedured for Performing Systematic Reviews," *Joint Technical Report TR/SE-0401*, Keele University, 2004.

[34] K. Petersen, S. Vakkalanka, and L. Kuzniarz, "Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update," *Information and Software Technology*, vol. 64, pp. 1–18, 2015. Available: https://doi.org/10.1016/j.infsof.2015.03.007

[35] H. Zhand, M. Ali Babar, and P. Tell, "Identifying Relevant Studies in Software Engineering," *Information and Software Technology*, vol. 53, no. 6, pp. 625–637, 2011. Available: https://doi.org/10.1016/j.infsof.2010.12.010

[36] M. Genero, A. Fernámdez-Saez, H. J. Nelson, G. Poels, and M. Piattini, "Research Review: A Systematic Literature Review on the Quality of UML Models," *Journal of Database Management*, vol. 22, no. 3, pp. 46–70, 2011. Available: https://doi.org/10.4018/jdm.2011070103

[37] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Weelén, *Experimentation in Software Engineering*. Springer, Heidelberg, 2012. Available: https://doi.org/10.1007/978-3-642-29044-2

[38] S. Abrahao, C. Gravino, E. Insfran, G. Scanniello, and G. Tortora: "Assessing the Effectiveness of Dynamic Modeling in the Comprehension of Software Requirements: Results from a Family of Five Experiments," *IEEE Transactions on Software Engineering*, vol. 39, no. 3, pp. 327–342, 2013. Available: https://doi.org/10.1109/tse.2012.27

[39] B. Anda, D Sjoberg, and M. Jorgensen, "Quality and Understandability of Use Case Models," *Proc. European Conference on Object-Oriented Programming, LNCS*, Springer, vol. 2072, pp. 402–428, 2001. Available: https://doi.org/10.1007/3-540-45337-7_21

[40] C. Amrit and N. Tax, "Towards Understanding the Understandability of UML Models," *Proc. 6th Intl. Workshop on Modeling in Software Engineering*, pp. 49–54, 2014.

[41] A. Maier, N. Baltsen, H. Christoffersen, and H. Störrle, "Towards Diagram Understanding: A Pilot Study Measuring Cognitive Workload Through Eye Tracking," *Proc Intl. Conference on Human Behavior in Design*, pp. 1–6, 2014.

[42] C. Bennett, J. Ryall, L. Spalteholz, and A. Gooch, "The Aesthetics of Graph Visualization," *Computational Aesthetics in Graphics, Visualization, and Imaging*, pp. 57–64, 2007. Available: https://doi.org/10.2312/COMPAESTH/COMPAESTH07/057-064

[43] J. Crus-Lemus, M. Genero, M. Manso, S. Morasca, and M. Piattini, "Assessing the Understandability of UML statechart diagrams with composite states – A family of empirical studies," *Empirical Software Engineering*, vol. 14, pp. 685–719, 2009. Available: https://doi.org/10.1007/s10664-009-9106-z

[44] J. Cruz-Lemus, M. Genero, and M. Piattini, "Using Controlled Experiments for Validating UML Statechart Diagram Measures," *Software Process and Product Measurement. Mensura IWSM 2007 2007. Lecture Notes in Computer Science*, Springer, vol. 4895, pp. 280–294, 2008. Available: https://doi.org/10.1007/978-3-540-85553-8_11

[45] J. Cruz-Lemus, M. Genero, M. Piattini, and A. Toval, "An Empirical Study of Nesting Level of Composite States Within UML Statechart Diagram," *Perspectives in Conceptual Modeling. ER 2005. Lecture Notes in Computer Science*, Springer, vol. 3770, pp.12–22, 2005. Available: https://doi.org/10.1007/11568346_3

[46] J. Cruz-Lemus, M. Genero, M. Esperanza Manso, M. Piattini, "Evaluating the Effect of Composite States on the Understandability of UML Statechart Diagrams," *Model Driven Engineering Languages and Systems, MODELS*

2005. Lecture Notes in Computer Science, Springer, vol. 3713, pp. 113–125, 2005. Available: https://doi.org/10.1007/11557432_9

[47] J. Cruz-Lemus, A. Maes, M. Genero, G. Poels, and M. Piattini, "The Impact of Structural Complexity on the Understandability of UML Statechart Diagrams," *Information Sciences*, vol. 180, no. 11, pp. 2209–2220, 2010. Available: https://doi.org/10.1016/j.ins.2010.01.026

[48] F. Di Cerbo. G. Dodero. G, Reggio. F. Ricca. and G. Scanniello, "Precise vs. Ultra-light Activity Diagrams – An Experimental Assessment in the Context of Business Process Modelling," *Product Focused Software Process Improvement, Lecture Notes in Computer Science*, Springer, vol. 6759, pp. 291–305, 2011. Available: https://doi.org/10.1007/978-3-642-21843-9_23

[49] K. Figl, "Symbol Choice and Memory of Visual Models," *Proc. IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 97–100, 2012. Available: https://doi.org/10.1109/vlhcc.2012.6344491

[50] M. Genero, D. Miranda, and M. Piattini, "Defining and Validating Metrics for UML Statechart Diagrams," *Proc. of the 6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering*, 2002.

[51] M. Genero, J. Olivas, M. Piattini, and F. Romero, "A controlled experiment for corroborating the usefulness of class diagram metrics at early phases of OO development," *Proc. of the 2nd ADIS 2001 Workshop on Decision Support in Software Engineering*, 2001.

[52] M. Genero, M. Piattini, and C. Calero, "Early Measure of UML Class Diagrams," *Obj. Logiciel Base données Réseaux*, vol. 6, no. 4, pp. 1–28, 2000.

[53] M. Genero, M. Piattini, and C. Calero, "Assurance of Conceptual Data Model Quality Based on Early Measures," *Proc. of 2nd Asia-Pacific Conference on Quality Software*, pp. 97–103, 2001. Available: https://doi.org/10.1109/apaqs.2001.990007

[54] M. Genero, M. Piattini, and C. Calero, "Empirical Validation of Class Diagram Metrics," *Proc. IEEE Intl. Symposium on Empirical Software Engineering*, pp. 195–203, 2002. Available: https://doi.org/10.1109/isese.2002.1166940

[55] M. Genero, G. Poels, and M. Piattini, "Defining and validating metrics for assessing the understandability of entity-relationship diagrams," *Data & Knowledge Engineering*, vol. 64, no. 3, pp. 534–557, 2008. Available: https://doi.org/10.1016/j.datak.2007.09.011

[56] W. Huang, "An Eye tracking Study into the Effects of Graph Layout," *preprint arXiv*:0810.4431, 2008.

[57] M. Heggest, J. Krogstie, and H. Wesenburg, "The Influence of Syntactic Quality of Enterprise Process Models on Model Comprehension," *Proc. of the CAiSE Forum at the 27th Intl. Conference on Advanced Information Systems Engineering*, 2015.

[58] J. Heiser and B. Tversky, "Arrows in Comprehending and Producing Mechanical Diagrams," *Cognitive Science*, vol. 30, no. 3, pp. 581–592, 2008. Available: https://doi.org/10.1207/s15516709cog0000_70

[59] J. Krogstie, "Integrating the Understanding of Quality in Requirements Specification and Conceptual Modeling. *ACM SIGSOFT Softw Eng Notes*, vol. 23, no. 1, pp. 86–91, 1998. Available: https://doi.org/10.1145/272263.272285

[60] L. Kuzniarz, M. Staron, and C. Wohlin, "An Empirical Study on Using Stereotypes to Improve Understanding of UML Models," *Proc. of the 12th IEEE Intl. Workshop on Program Comprehension*, 2004. Available: https://doi.org/10.1109/wpc.2004.1311043

[61] K. Lemon, E. Allen, J. Carver, and G. Bradshaw, "An Empirical Study on the Effects of Gestalt Principles on Diagram Understandability," *Proc. of the 1st Intl. Symposium on Empirical Software Engineering and Measurement*, 2007. Available: https://doi.org/10.1109/esem.2007.37

[62] C. Lange, B. DuBois, M. Chaudron, and S. Demeyer, "Experimentally Investigating the Effectiveness and Effort of Modeling Conventions for the UML," *Proc. of the 10th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering*, 2006.

[63] M. Manso, J. Cruz-Lemus, M. Genero, and M. Piattini, "Empirical Measures for UML Class Diagrams: A Meta-Analysis Study," *Models in Software Engineering. MODELS 2008. Lecture Notes in Computer Science*, Springer, vol. 5421, pp. 303–313, 2009. Available: https://doi.org/10.1007/978-3-642-01648-6_32

[64] M. Marchesi, "OOA metrics for the unified modeling languages," *Proc. 2nd Euromicro Conf. Software Maintenance and Reengineering*, pp. 67–73, 1998. Available: https://doi.org/10.1109/csmr.1998.665739

[65] K. Marriott, H. Purchase, M. Wybrow, and C. Goncu, "Memorability of Visual Features in Network Diagrams," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2477–2485, 2002. Available: https://doi.org/10.1109/tvcg.2012.245

[66] A. Nugroho, "Level of Detail in UML Models and its impact on Model comprehension: A controlled Experiment," *Information and Software Technology*, vol. 51, no. 12, pp.1670–1685, 2009. Available: https://doi.org/10.1016/j.infsof.2009.04.007

[67] A. Nysetvold and J. Krogstie, "Assessing Business Process Modeling Languages Using a Generic Quality Framework," *Advanced Topics in Database Research*, vol. 5, pp 79–93, 2006. Available: https://doi.org/10.4018/978-1-59140-935-9.ch005

[68] H. Purchase, L. Colpoys, M. McGill, D. Carrington, and C. Britton, "UML class diagram syntax: An empirical study of comprehension," *Proc. of the Australian Symposium on Information Visualization*, 2001.
[69] G. Porras and Y. Gueheneuc, "An Empirical Study on the Efficiency of Different Design Pattern Representations in UML Class Diagrams," *Empirical Software Engineering*, vol. 15, no. 5, pp. 493–522, 2010. Available: https://doi.org/10.1007/s10664-009-9125-9
[70] H. Purchase, M. McGill, L. Colpoys, and D. Carrington, "Graph drawing aesthetics and the comprehension of UML class diagrams: an empirical study," *Proc. of the 2001 Asia-Pacific Symposium on Information Visualisation*, vol. 9, pp. 129–137, 2001.
[71] J. Recker, M. Muehlen, K. Siau, J. Erikson, and M. Indulska, "Measuring Method Complexity: UML versus BPMN," *Proc. of the 15th Americas Conference on Information Systems*, pp. 1–9, 2009.
[72] F. Ricca, M. Penta, M. Torchiano, P. Tonella, and M. Ceccato, "The Role of Experience and Ability in Comprehension Tasks supported by UML Stereotypes," *Proc. of the 29th International Conference on Software Engineering*, pp. 375–384, 2007. Available: https://doi.org/10.1109/icse.2007.86
[73] G. Reggio, F. Ricca, G. Scanniello, F. Cerbo, and G. Dodero, "A Precise Style for Business Process Modelling: Results from Two Controlled Experiments," *Proc. of the Intl. Conf. on Model Driven Engineering Languages and Systems*, pp 138–152, 2012. Available: https://doi.org/10.1007/978-3-642-24485-8_11
[74] H. Störrle, "On the Impact of Layout Quality to Understanding UML Diagrams," *Proc of the IEEE Symp. Visual Languages and Human-Centric Computing*, pp. 135–142, 2011. Available: https://doi.org/10.1109/vlhcc.2011.6070390
[75] L. Sanchez-Gonzlez, F. Garcia, J. Mendling, F. Ruiz, and M. Piattini, "Prediction of Business Process Model Quality based on Structural Metrics," *Proc. of the Intl. Conf. on Conceptual Modeling*, pp. 458–463, 2010. Available: https://doi.org/10.1007/978-3-642-16373-9_35
[76] M. Staron, L. Kuzniarz, and C. Wohlin, "Empirical Assessment of Using Stereotypes to Improve Comprehension of UML Models: A Set of Experiments," *Journal of Systems and Software*, vol. 79, no. 5, pp 727–742, 2006. Available: https://doi.org/10.1016/j.jss.2005.09.014
[77] B. Sharif and J. Maletic, "An Empirical Study on the Comprehension of Stereotyped UML Class Diagram Layouts," *Proc. of the 17th IEEE Intl. Conference on Program Comprehension*, pp. 268–272, 2009. Available: https://doi.org/10.1109/icpc.2009.5090055
[78] B. Sharif and J. Maletic, "An Eye Tracking Study on the Effects of Layout in Understanding the Role of Design Patterns," *Proc. of the IEEE Intl. Conference on Software Maintenance*, pp. 1–10, 2010. Available: https://doi.org/10.1109/icsm.2010.5609582
[79] B. Sharif and J. Maletic, "The Effect of Layout on the Comprehension of UML Class Diagrams: A Controlled Experiment," *Proc. of the 5th IEEE Intl. Workshop on Software Visualization*, pp. 11–18, 2009. Available: https://doi.org/10.1109/vissof.2009.5336430
[80] T. Wahl and G. Sindre, "An Analytical Evaluation of BPMN Using Semiotic Quality Framework," *Advanced Topics in Database Research*, vol. 5, pp. 94–105, 2006. Available: https://doi.org/10.4018/978-1-59140-935-9.ch006
[81] S. Yusuf, H. Kagdi, and J. Maletic, "Assessing the Comprehension of UML Class Diagrams via Eye Tracking," *Proc. of the 15th IEEE Intl. Conf. on Program Comprehension*, pp. 113–122, 2007. Available: https://doi.org/10.1109/icpc.2007.10
[82] T. Yi, F. Wu, and C. Gan, "A Comparison of Metrics for UML Class Diagrams," *ACM SIGSOFT Software Engineering Notes*, vol. 29, no. 5, pp. 1–6, 2004. Available: https://doi.org/10.1145/1022494.1022523
[83] S. Zugal, J. Pinggera, B. Weber, J. Mendling, and H. Reijers, "Assessing the Impact of Hierarchy on Model Understandabiliy – A Cognitive Perspective," in J. Kienzle (eds) *Models in Software Engineering. MODELS 2011. Lecture Notes in Computer Science*, Springer, vol. 7167, pp. 123–133, 2011. Available: https://doi.org/10.1007/978-3-642-29645-1_14
[84] S. Zugal, P. Soffer, J. Pinggera, and B. Weber, "Expressiveness and Understandability Considerations of Hierarchy in Declarative Business Process Models," *Enterprise, Business-Process and Information Systems Modeling. BPMDS EMMSAD 2012. Lecture Notes in Business Information Processing*, Springer, vol. 113, pp. 167–181, 2012. Available: https://doi.org/10.1007/978-3-642-31072-0_12