On the impact of size to the understanding of UML diagrams

Störrle, Harald

Published in:
Software and Systems Modeling

Link to article, DOI:
10.1007/s10270-016-0529-x

Publication date:
2016

Document Version
Peer reviewed version

Citation (APA):
Störrle, H. (2016). On the impact of size to the understanding of UML diagrams. *Software and Systems Modeling*, *17*(1), 115-134. https://doi.org/10.1007/s10270-016-0529-x

J. Software and Systems Modeling (SoSyM) manuscript No.
(will be inserted by the editor)

On the Impact of Size to the Understanding of UML Diagrams

Harald Störrle

Received: May 6th, 2015, revised: December 12th, 2015, final: April 20th, 2016

Abstract
Background: Practical experience suggests that usage and understanding of UML diagrams is greatly affected by the quality of their layout. While existing research failed to provide conclusive and comprehensive evidence in support of this hypothesis, our own previous work provided substantial evidence to this effect, also suggesting diagram size as a relevant factor, for a range of diagram types and layouts.
Aims: Since there is no generally accepted precise notion of "diagram size", we first need to operationalize this concept, analyze its impact on diagram understanding, and derive practical advice from our findings.
Method: We define three alternative, plausible metrics. Since they are all highly correlated on a large sample of UML diagrams, we opt for the simplest one. We use it to re-analyze existing experimental data on diagram understanding.
Results: We find a strong negative correlation between diagram size and modeler performance. Our results are statistically highly significant, and exhibit a very large degree of validity. We utilize these results to derive a recommendation on diagram sizes that are, on average, optimal for model understanding. These recommendations are implemented in a plug-in to a widely used modeling tool, providing continuous feedback about diagram size to modelers.
Conclusions: The effect sizes are varying, but generally suggest that the impact of size matches or exceeds that of other factors in diagram understanding. With the guideline and tool, modelers are steered towards avoiding too large diagrams.

Keywords Diagram Understanding · Diagram Size Metrics · Cognitive Load · Experiment · Gestalt Principles

H. Störrle
Dept. of Applied Mathematics and Computer Science
Technical University of Denmark
E-mail: hsto@dtu.dk

1 Introduction

The Unified Modeling Language (UML) has been called the "*lingua franca of software engineering*" [39, p. vi] for over 15 years now. It is a generally held belief that visual languages are superior to textual languages in that they support human perceptual and thought processes, and that this is also true for the UML, in fact, that this is a major reason for the adoption and success of UML. However, there are actually no compelling research results to support this belief. There are theoretical works such as [14, 23] who point out the importance of layout and provide elements of a conceptual framework for studying layouts, but do not provide empirical evidence. There is also a body of experimental results on the layout of UML class diagrams and how it affects human understanding and problem solving. However, there are substantial differences between the layouts of class diagrams and most of the 13 other notations proposed by the UML, so the findings lack generalizability. Also, most of the findings are ambiguous or inconclusive, and sometimes unintuitive. In particular, only very small effects have been found in vitro. For instance, Eichelberger and Schmid note that "*We could not identify [...] a significant impact [by diagram quality]*" (cf. [9, p. 1696]).

On the other hand, practical experience in industrial software projects suggests a much higher impact of good or bad layout, and previous work by the author strongly supports this hypothesis (see [47, 48]). Inspection of our data and a qualitative study with our study participants suggested, however, that the size of the models portrayed in the diagrams might be a relevant factor. In order to study this question, we define a precise notion of diagram size and re-examine existing data sets of substantial size (78 participants, well over 1200 measurements). Our working hypothesis is that modeler performance correlates negatively with diagram size. We also hypothesize, that layout quality matters more with increasing diagram size: small diagrams are easy to use irrespective of the layout quality simply because they are small; modelers simply cope with bad layout. With increasing diagram size, however, the visual and/or mental capacity of a modeler is stretched, so that the impact of poor layout quality to modeler performance will show up more and more. In other words, layout quality matters more, and is more apparent for larger diagrams. Note that, while we will elaborate on the notion of layout quality in Section 2 below, a precise and quantitative definition of this notion is beyond the scope of the present paper. For the time being, the terms "good" and "bad" will remain intuitive—exploring the notion of diagram size is a necessary but insufficient step towards a more satisfying definition of *diagram quality*.

In our previous publications [47,48] we have analyzed the impact of UML diagram layout quality to various indicators of modeler performance including score and errors, preference and assessment, and cognitive load (cf. [25]). We could show that layout quality does indeed have a major impact on the understandability of diagrams in all of these dimensions. We found that this holds irrespective of diagram type, but is modulated by expertise level. In the process of the analysis, the data also suggested a strong correlation between diagram size and outcome, based on a tentative and informal notion of diagram size. So we conducted the present follow-up study that defined this informal notion in a precise way, and studied the connection to modeler performance.

In order to analyze the relationship between diagram size and modeler performance, we need to formalize the notion of diagram size first, since such a concept did not exist. We have defined three plausible, progressively refined metrics of diagram size and calculated them on the 36 diagrams previously studied in [48]. Surprisingly, there is a high correlation between all the metrics, so, by Occam's razor, we selected the simplest one, and have correlated it to the previously measured outcomes. We could identify a strong negative correlation between diagram size and modeler understanding of the respective diagram.

Exploiting this relationship in the opposite direction, we can determine practical limits to the size of diagrams that afford being understood easily and correctly by modelers. This limit is useful as a guideline to inexperienced modelers, such as students. We have also implemented these guidelines in a plug-in to a UML modeling tool so that modelers receive continuous feedback about the size of their diagrams.

Paper outline and relationship to previous work We first describe the commonly accepted criteria for good diagrams (Section 2). We derive diagram size metrics and compare them (Section 3), and explain the experiments conducted (Section 4). The analysis of the factor "diagram size" (Section 5) are presented, and threats to validity are discussed (Section 6). We also implement and validate our work as a tool for diagram size monitoring and provide some feedback on its usage by students (Section 7). Section 8 discusses the related work, focusing particularly to the more recent publications and broadening the scope considerably. Finally, we summarize our findings, assess their contribution to the state of the art, and outline future work (Section 9).

This paper is a much extended version of [49], adding large parts of Section 3, elaborating with more a detailed analysis and more data in Sections 4 and 5, providing the tool the implementation and validation (Section 7), and also comprehensively updating the Related Work (Section 8).

2 Quality of Diagram Layout

In this section we summarize the well-known rules of good layout, as far as they are relevant to UML and similar diagrams. We take these as a given, leading to a mostly intuitive understanding of good and bad layout. Rather than trying to formally define these vague notions directly, we shall first retreat to the simpler and more basic notion of size, thus focusing on objective aspects of diagrams. In future work, this notion can then be used for a better definition and study of the complexity and quality of diagram layout.

2.1 Diagram layout levels

In this section, we will briefly review the knowledge on aesthetic criteria for the layout of UML diagrams and its effects on model understanding. A detailed discussion of aesthetic criteria for class diagrams is found in [7, p. 54–65], a recent survey of empirical results on layout criteria is found in [9]. Wong and Sun [54] provide an overview of these criteria from a cognitive psychology point of view, along with an evaluation of how well these principles are realized in several UML CASE tools. Purchase et al. discuss aesthetic criteria with a view to the layout of UML class and communication diagrams (cf. [32,31]) and also provide sources to justify and explain these criteria (cf. [29]). Eichelberger [6] also discusses these criteria at length, and shows how they can be used in the automatic layout of UML class diagrams.

The layout of UML diagrams is governed by four levels of design principles (see Fig. 2). First, there are the general principles of graphical design and visualization that apply to all kinds of diagrams, and probably any kind of visualization. For instance, in a good layout, elements should not obscure each other, the Gestalt principles should be respected [16], text should be shown in a readable size, elements should be aligned (e.g., on a grid), and there should be sparing and careful use of colors, and different fonts or styles. This is the "grapheme" level [11] that is addressed by a theory like Moody's "Physics of Notation" [22,51].

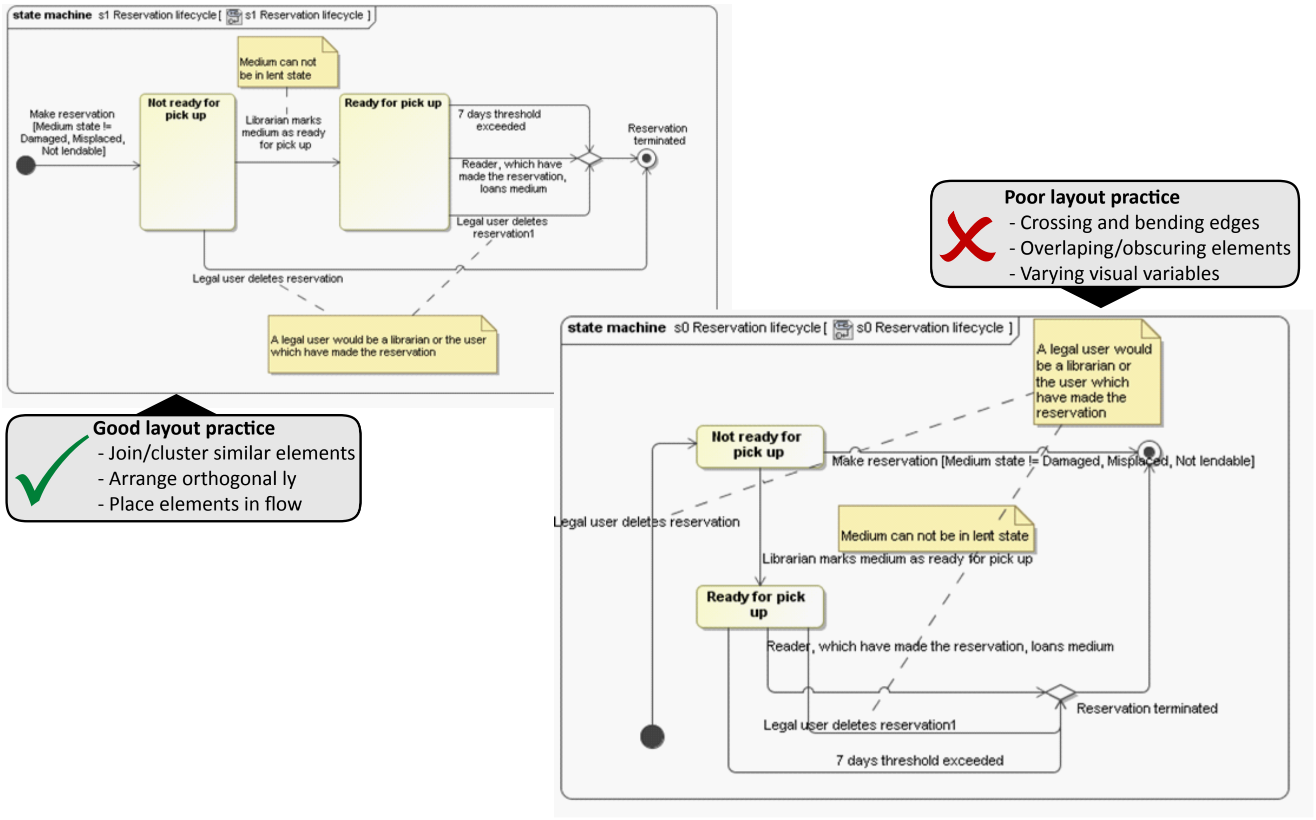

state machine s1 Reservation lifecycle [s1 Reservation lifecycle]
Medium can not be in lent state
Make reservation [Medium state != Damaged, Misplaced, Not lendable]
Not ready for pick up
Librarian marks medium as ready for pick up
Ready for pick up
7 days threshold exceeded
Reader, which have made the reservation, loans medium
Legal user deletes reservation1
Reservation terminated
Legal user deletes reservation
A legal user would be a librarian or the user which have made the reservation
Good layout practice
- Join/cluster similar elements
- Arrange orthogonal ly
- Place elements in flow
Poor layout practice
- Crossing and bending edges
- Overlaping/obscuring elements
- Varying visual variables
state machine s0 Reservation lifecycle [s0 Reservation lifecycle]
A legal user would be a librarian or the user which have made the reservation
Not ready for pick up
Make reservation [Medium state != Damaged, Misplaced, Not lendable]
Legal user deletes reservation
Medium can not be in lent state
Librarian marks medium as ready for pick up
Ready for pick up
Reader, which have made the reservation, loans medium
Reservation terminated
Legal user deletes reservation1
7 days threshold exceeded

Fig. 1 Examples of good/bad layouts of a diagram as used in the study

Second, there are layout principles applying to all structures that can be considered a graph in the mathematical sense. Thus, good layouts should avoid or minimize crossings, bends, and length of lines. Most of the empirical research on UML diagrams focuses on principles from this level, for instance the work by Purchase, Eichelberger, Maletic, Sharif, and others (see Section 8 for a more detailed account).

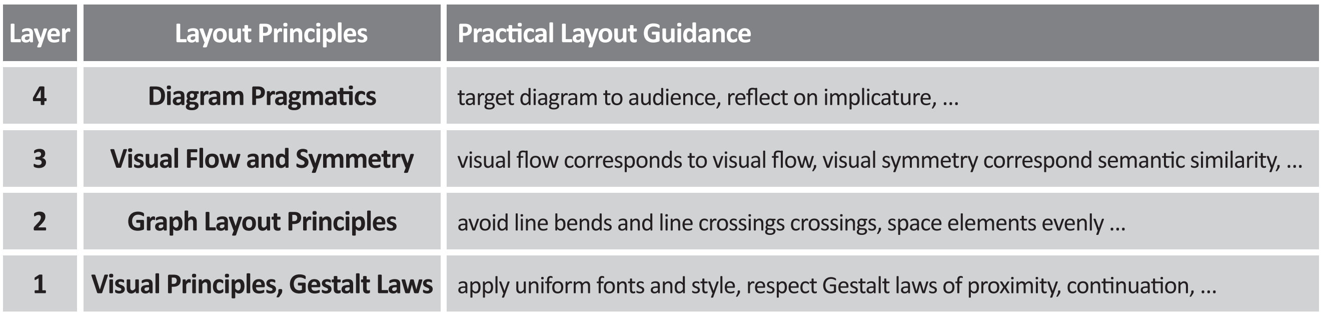
Third, there are layout principles that apply mostly only to notations like those found in UML. For instance, diagrams with some inherent ordering of elements should maintain and highlight that ordering as visual flow. Visual clutter should be reduced by introducing symmetry when possible. For instance, similar edges should be joined, similar elements should be aligned and grouped, and so on. In UML, this means that if a class has several subclasses, it might be helpful to group and align the subclasses and join the arcs indicating the inheritance-relationship. Another application is found in activity diagrams, where several consequences of a decision could be aligned and grouped. This might be called the level of layout patterns, which has so far not been studied in great detail.

Fourth, there is the level of pragmatics, that is, support for underlining the purpose of a diagram in order to better address the audience. Items may be highlighted by color, size, or position to guide and direct the attention of readers. On this level, rules and guidelines from lower levels may be put aside to better serve the paramount purpose of conveying the message and telling whatever story the diagram designer intends to tell. This level has been called pragmatic [23, 14].

2.2 Diagram layout factors

Previous research was motivated by the rationale of improving automatic layout algorithms by finding weighing factors that would result in layouts that appeal more to humans. In that sense, the point of the research was not primarily to study human behavior and perception, but the tools and algorithms they intended to calibrate. As a consequence, most previous studies focused on individual low-level layout principles of the first and second level mentioned above (see Fig. 2, and cf. [38, 7, 10, 55, 32]).

The results show that there are many factors for the (perceived) quality of diagram layout, and that their impact varies. All of these factors, though, seem to have a rather small impact individually, as existing results had only marginal statistical significance, and pointed to small effect sizes (see e.g. [32, 31, 29]). We believe that this may be a consequence of the rather small populations, and the specific stimulus samples used in experiments by Purchase and others.



Layer	Layout Principles	Practical Layout Guidance
4	**Diagram Pragmatics**	target diagram to audience, reflect on implicature, …
3	**Visual Flow and Symmetry**	visual flow corresponds to visual flow, visual symmetry correspond semantic similarity, …
2	**Graph Layout Principles**	avoid line bends and line crossings crossings, space elements evenly …
1	**Visual Principles, Gestalt Laws**	apply uniform fonts and style, respect Gestalt laws of proximity, continuation, …

Fig. 2 The principles of diagram layout are organized into four layers.

Also, the ranking and contribution of these criteria vary across different diagram types. Even between class and communication diagrams, which are rather close relatives as far as concrete syntax and layout are concerned, Purchase et al. show notable differences in the ordering and impact of layout criteria [32, pp. 246]. While this question has not been explored for other diagram types, it is likely they will exhibit an even greater variance, in particular regarding notations with greater conceptual and visual differences. For instance, flow is clearly a dominating visual property of Activity diagrams, whereas it is much less prominent in Class diagrams, and yet another visual style is found in Sequence diagrams. Intuitively, different diagram types constitute different visual languages that afford different visual styles, implying different criteria for the assessment and optimization of layout. This observations is well captured by the four layers of layout we outlined in the previous section: we sometimes optimize for higher level principles (such as overall diagram flow) at the expense of lower level flaws (such as line bends and crossings). Therefore, we conjecture that higher level layout principles may overrule lower level principles, and that good layouts can only be created with a holistic approach including all layers simultaneously, thus taking also into account the particularities of the respective visual language. In other words, there is no single layout algorithm and set of calibration factors across visual notations.

For humans creating diagram layouts, however, a set of comparatively vague guidelines together with some instruction is often good enough for practical purposes. Humans may (and will) mix and match criteria from all four levels as appropriate and create what they *and their peers* perceive as high quality UML diagrams. Of course, there is still a large degree of subjectivity in this definition, but it does capture the intuition.[1] Therefore, in the remainder of this paper, we will call a diagram (layout) *good*, if it (mostly) adheres to the criteria from all these levels, and *bad* if it violates them. Generally speaking, in terms of the four levels of layout rules described above, if a diagram layout does not (significantly) violate any of the rules on the first two levels but (more or less) adopts the rules described in the latter two levels we call it a "good" layout. Conversely, we call a diagram layout "bad" if it consistently violates these rules. An more precise definition of "good" and "bad" layout is not possible at this point, and beyond the scope of this paper. The evidence provided below, however, takes us closer to such an objective, quantifiable definition of the quality of diagram layout.

3 Size of UML diagrams

Surprisingly, there are no generally accepted metrics on model size, and apparently none at all for diagram size, neither on the context of UML nor any other similar modeling notation. Thus, we are lacking a reference point for the correlation to modeler performance and need to define such a metric first. We will visit three plausible candidates and compare them to find the most appropriate.

3.1 Metric 1: Number of Diagram Elements (NODE)

We believe the simplest conceivable metric is to simply count the number of diagram elements[2], giving the NODE metric.

$$\textbf{(NODE)} \qquad size_{\mathrm{N}}(d) := \mid \{element \in d\} \mid$$

Following the argument presented in [51], we assume that there are effectively three different kinds of graphemes: lines, icons, and shapes, where the latter can be further split into simple shapes like geometric figures, and complex shapes that made up of other graphemes (see Fig. 3).

The justification for treating icons such as the stick figure representing actors differently than complex shapes is derived from perceptual psychology, in particular Gestalt

[1] Observe that the diagrams in the "good layout" treatment of this and previous studies were optimized by hand, and [47,48] shows that the optimization was indeed successful.

[2] Diagram elements are not to be confused with the model elements shown in the diagram.

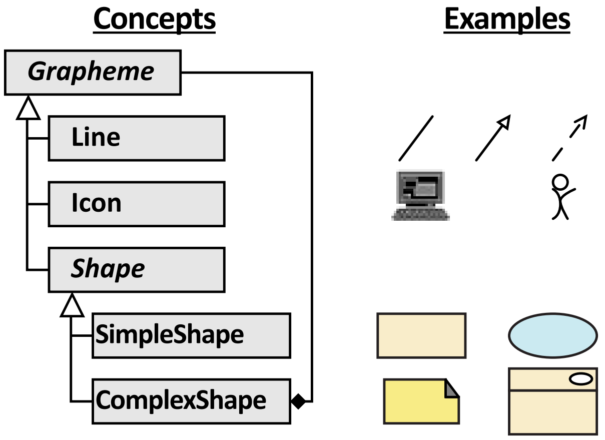


Fig. 3 Kinds of shapes considered by our metrics.

psychology[3] Clearly, the elements of icons are very close to each other, so much so, that by the Gestalt law of proximity, they are perceived as belonging together. Also, the brain abstracts semantically connected elements into larger, more semantic units, an operation known as chunking, in order to improve the utilization of short term memory [21]. Thus, we hypothesize, that e.g. a stick figure is processed and perceived as a unit rather than a compound.

Based on the Gestalt laws [56], and common sense about modeling notations, we defined counting rules as follows. We count every element of a diagram that has discernible visual features, such as rectangles representing classes, ellipses representing use cases, straight lines representing associations, and so on. Regarding bended or multi-legged lines, we count each segment individually, but we include integral parts and adornments such as arrow heads, attached names, or stereotypes required for element disambiguation.[4] We count textual labels as additional individual elements if and only if they are not a necessary part of an element. For instance, the label representing the name of a class is considered to be included in the count of the rectangle. However, properties of the class are counted as one label each. Consider the diagrams shown in Fig. 4. Both of these diagrams contain 3 shapes, 4 line segments, and 1 label. The following table shows the counts for the other examples shown in Figures 1 and 12.

DIAGRAM	SHAPES	LINE SEGMENTS	TEXTUAL LABELS	SUM
Fig. 1 (left)	7	14	7	28
Fig. 1 (right)	7	17	7	31
Fig. 12	18	25	20	63

3.2 Metric 2: Weighted Number of Diagram Elements (WNODE)

Obviously, the NODE metric is trivial to define and straightforward to compute. However, it is arguably not just simple, but too simple. For one thing, it disregards topological information (i.e., containment), which certainly contributes to diagram complexity and information content. Also, NODE does not take into account differences in complexity among the potential elements of a diagram. Clearly, diagram elements come with varying degrees of details, and thus convey different amounts of information to the reader. Rather than counting all elements per se, we might want to capture their contribution in terms of complexity or information content. The simplest way of doing this is to include a weight factor for the individual types of elements which compensates for differences between element types. However, it is not quite clear what the "right" weights are, and how we may obtain them. In order to decide this, we need to explore the decomposition of graphemes.

3.2.1 Decomposition of complex graphemes

Many of the graphemes defined by the UML are compounds, that is, they are made up of other, simpler graphemes. Inevitably, this gives rise to alternative decompositions. For instance, a class with a name and two compartments may be interpreted as one big rectangle with a label and two lines, or as one rectangle with a label and two adjacent rectangles of same width (see Fig. 5).

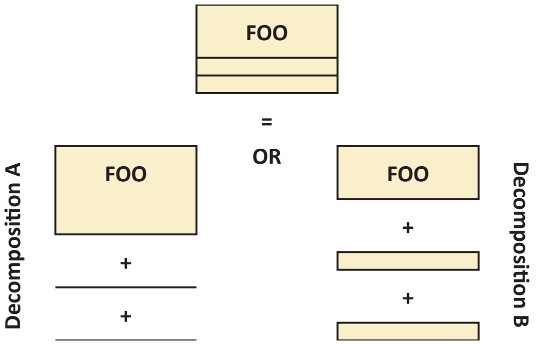


Fig. 5 Even the most pedestrian of UMLs graphemes give rise to alternative decompositions: a class with two compartments may be interpreted as one big rectangle with two lines, or as three adjacent rectangles of equal width.

For more complex examples, even more decompositions arise. Consider a UseCase with an extension point represented as a Class with an attached visual stereotype (a notation proposed in [24, p. 675]). Fig. 6 illustrates this example. It could be either parsed into a rectangle with label, a line, a small ellipse, and two grouped labels, or it could be decomposed into two rectangles, one with a ("built in") label and

[3] The Gestalt laws have first been described in the 1920s by Köhler [17] and others; they are today covered by any introductory textbook on perceptual and cognitive psychology. Consult [16] for one of the original sources.

[4] Observe that this way, we organically include curved lines which are defined by a set of auxiliary points just like multi-legged lines.

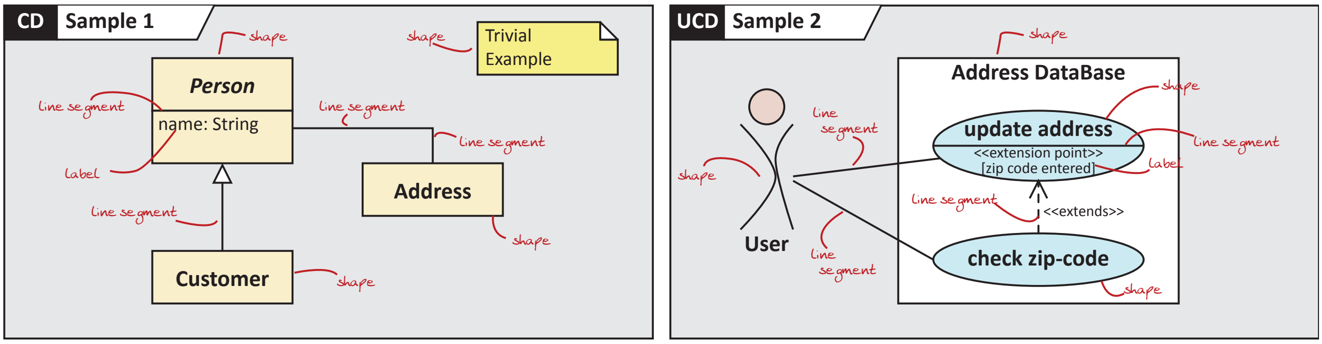


Fig. 4 Two simple examples of counting diagram elements: a class diagram (left) and a use case diagram (right), both of which contain 4 shapes, 4 line segments, and 1 label, totaling 9 elements. Observe that (most) shapes include one label, as do lines where the label is indispensable to disambiguate the meaning of the line, such as the stereotypes in the use case diagram.

a small ellipse, the other with two amalgamated labels. The same problem occurs with other notations in the UML, and in fact, many other notations beyond UML.

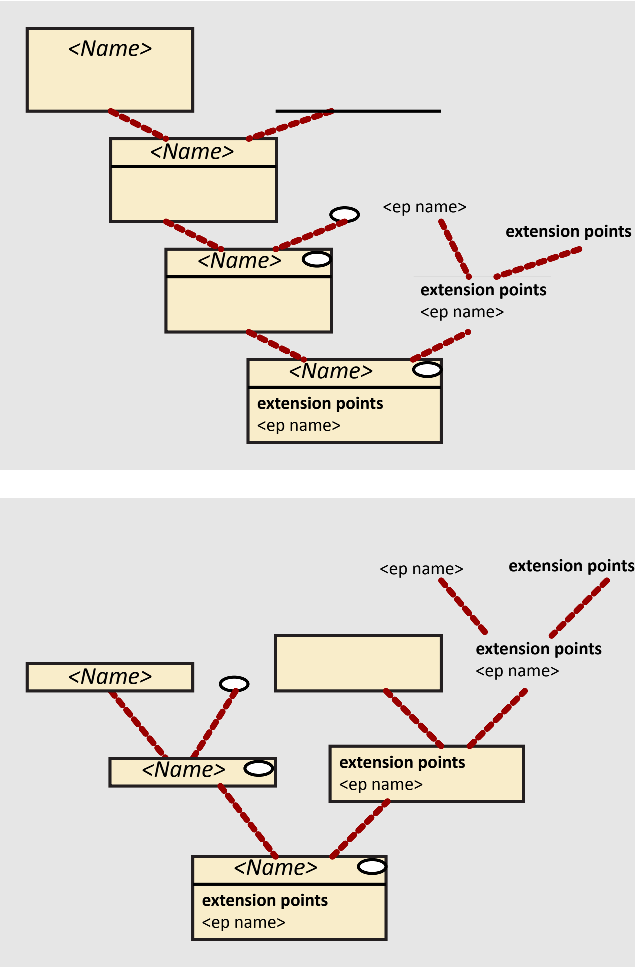


Fig. 6 Decomposition of graphemes is ambiguous: of these two alternative decompositions, the Gestalt laws suggest that the decomposition at the top is more commonly found as the “natural” interpretation.

Perceptual psychology has shown that the human visual apparatus prefers certain interpretations over others, as triggered by particular geometric cues. Generally speaking, the “natural” interpretations are simpler or less complex than other, theoretically equally possible interpretations, leading to a plethora of visual paradoxes. The findings of [53] suggest that it is mostly the larger and/or enclosing structure that is decisive for the overall interpretation of compound shapes. We emulate these findings by parsing “outside in”, and deciding for the smaller overall element count when decomposing compound graphemes.

3.2.2 Relative weights of graphical primitives

We postulate that the different groups of graphical primitives we have identified above have different weight in the sense that they impose varying degrees of cognitive load on modelers. The exact values of these weights can only be determined experimentally, so we leave them parametric for the time being, defining a generic weight factor $weight(e)$ for diagram elements e, that depends on the element type and its complexity level. All in all, we can now define the visual size of a UML diagram as

$$\textbf{(WNODE)} \qquad size_W(d) := \sum_{e \in d} weight(e)$$

where we use the notation $e \in d$ to indicate that e is an element of diagram d. We make the following judgments.

- Decorations at the beginning or end of a line or line segment (such as arrow heads) are considered to be an integral part of the line (as before), but increase its complexity.
- Simple shapes are considered of low complexity, shapes containing other shapes are considered to be of medium complexity, while complex shapes and icons are considered to have high complexity. The name-label of a shape is included if its complexity does not exceed that of the underlying shape; in that case, the complexity of the label prevails.

- Labels are strings of text that are attached to or positioned relative to other elements. Labels are restricted to single lines. Single characters or short names are considered simple, long names are considered as medium complex, and structured expressions like sentences or operation declarations are considered to be highly complex.

With these conventions, we define diagram size as the number of elements in a diagram, weighted by their complexity (e.g., one might define the weights 1, 1.5, and 2 for low, medium, and high complexity). This metric is substantially more difficult to compute than NODE, but it reflects the intuition more accurately, and could thus be expected to be more realistic, and provide higher validity.

3.3 Metric 3: Adjusted Number of Diagram Elements (ANODE)

Still, one might argue that the second approach is too simplistic, as the influence of diagram types is not considered. After all, every UML diagram establishes a context that restricts the admissible vocabulary in this diagram to a small subset of the overall UML meta-model. The vocabularies can differ significantly by diagram type. For instance, there are many more notational elements in the UML sub-language of Activity Diagrams than there are in the sub-language of Use Case Diagrams. Thus, according to classic information theory, the weight of any element in an Activity Diagram ought to be higher than the weight of the elements in Use Case Diagrams. That way, more information is conveyed by an Activity Diagram than by a Class Diagram of the same size.

In analogy with classic information theory, the number of choices should determine the information content (i.e., the weight) of a diagram element. We compute the information content of diagram elements as the binary logarithm of the set of similar elements a modeler may chose from, per diagram type. So, for every diagram element e from a class E of diagram elements in a given diagram type, we compute the weight of an element as the logarithm of the vocabulary size. Using this as a weight factor provides a third metric of diagram size.

$$\textbf{(ANODE)} \qquad size_A(d) := \sum_{e \in d} log_2 |E_d|$$

where E_d is the set of admissible elements in the notation used to express d. we use the notation $e \in d$ to indicate that e is an element of diagram d. Observe that there is no difference between NODE and ANODE when looking at one single diagram type: any observable effect would appear equally in both metrics. However, when comparing different types of diagrams, differences ought to become visible.

3.4 Comparing NODE, WNODE, and ANODE

Applying the three metrics defined above to compute the size of the diagrams given in Fig. 1 yields the values 30, 27.7, and 48.1 for the good layout and 34, 29.9, and 52.1 for the bad layout, respectively. Obviously, these metrics assign different size values to the different diagrams even though the diagrams represent the same model and, in some sense, do not convey different amounts of information. However, the poor arrangement adds line segments, so that a modeler is in fact dealing with more information. In that sense, all the above metrics satisfy their purpose.

Clearly, we will need to validate these diagram size metrics. So, we computed the sizes according to each measure with some (sensible) variations for the weights of the second metric for the same 38 diagrams that have been used in [47,48]. We compared the outcomes pairwise using Pearson's product-moment correlation. Surprisingly, we found that all three measures show very high levels of correlation with each other (0.967, 0.983, and 0.992, respectively) with very high confidence ($p < 10^{-15}$). That is to say: the measures do not yield significantly different results, it does not matter which metric we use. So, by Occam's razor, we decide for the one that offers the practical advantage of being simple to compute, that is, in the remainder we simply count the number of diagram elements as a metric for diagram size (NODE).

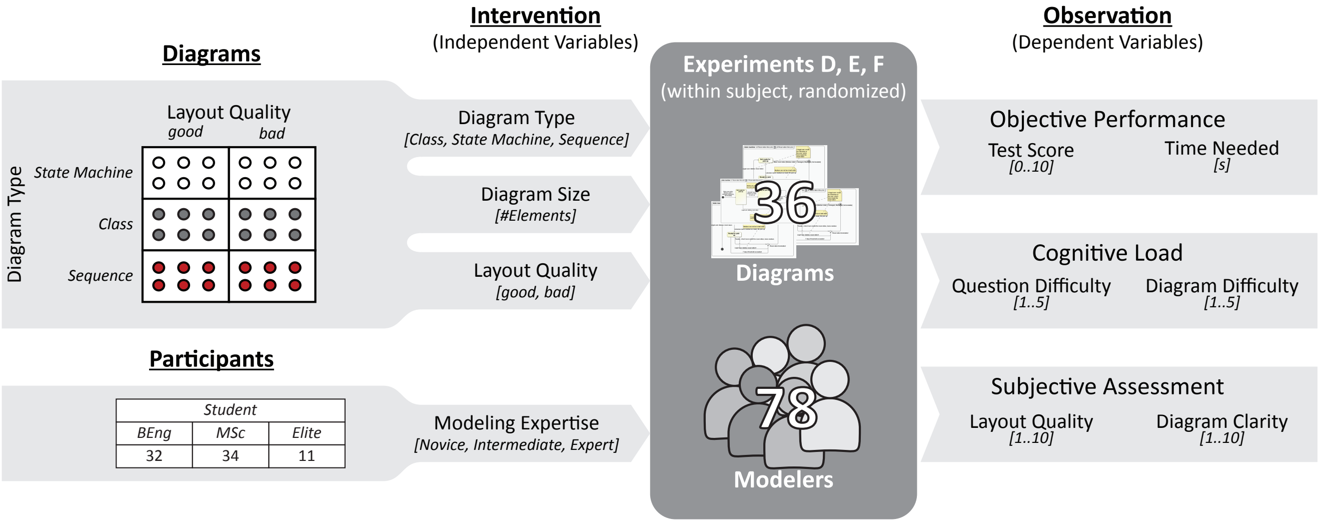
4 Experimental setup

The data which we analyze in this paper has been obtained by a series of experiments [48]. We restrict ourselves here to a cursory description of that experiment. We have kept the terminology and identifiers used in the earlier publication to allow easier comparison, at the price of some unobvious names in this paper.

The experiments were conducted on three disjoint populations of students at different levels of expertise. Students were given a set of sheets where each sheet contained a UML diagram, ten questions about the model visualized by the diagram, and questions to assess the difficulty and clarity of the diagram, personal preference, and subjective assessment of layout quality. Demographic data was collected along with an informed consent sheet.

The experiment was designed using [27] as a guideline. The dependent variables are accuracy and speed of comprehension, and preference. The independent variables are population, diagram type, and layout quality as measured following or violating the characterization of layout quality given in Section 2. This setup is visualized in Fig. 7.

In total, 78 students participated (completion rate over 80%), each answering questions for up to 9 out of 36 diagrams. The diagrams were extracted from three different



Student		
BEng	MSc	Elite
32	34	11

Fig. 7 Outline of the experimental setup: independent variables (left), study parameters (middle), and dependent variables (right).

case studies to ensure participants could not carry over experience from one diagram to another. The sequences of diagrams presented were counterbalanced to eliminate learning effects, resulting in four different questionnaires that were randomly assigned to participants. Instructions were carefully created and provided in writing to reduce experimenter bias.

4.1 Model population

The models used in the experiments have been created by students as part of their coursework in a requirements engineering course taught by the author. These models belonged to one of three case studies and have been prepared by teams of 4-7 students over a period of twelve weeks with an approximate effort of 600-800 working hours for each model. For each case study, two or three teams worked in parallel; for each case study, the model of the team achieving the highest grade was selected. This procedure ensured several desirable properties.

Firstly, by using models created by students undergoing the same course and being awarded the same grade, very similar levels of modeler capability and model quality may be assumed. Furthermore, the models used exhibit a large degree of methodological homogeneity in that they are very similar in terms of model structure and size, model and diagram usage, and frequency distribution of diagram types. Also, in the models used in our experiments, model elements had their original, semantic-bearing names, whereas in some previous experiments this vital aspect seems to have been deliberately eliminated by giving meaningless synthetic names to model elements (cf. [9, p. 1697]). Secondly, the course is evaluated by practitioners rather than academics, and the evaluation employs realistic evaluation criteria. Therefore, we feel justified in claiming that the models underlying our experiment are realistic wrt. their size, quality, and purpose are very close to industrial reality. Finally, all of these models used exist at the same stage of the software life cycle, namely requirements analysis.

In contrast, all earlier works seem to have used only a single case study and model, and most work has been carried out on models at the design or implementation level. Also, there is no indication in previous work as to how close to the reality of practical software development the underlying models are.

We have also analyzed our model sample for size by type (Fig. 8, top). The differences are not statistically significant. For comparison, we have estimated the size of the class diagrams stored in the Gothenburg UML repository (GUR, see [15]). We found that sizes range from 2 to 314 elements (median 39, mean 45.54, std. dev.=33.34, n=810), with the middle quartiles ranging fro 22 to 59 elements.[5] The distribution of the GUR sample is shown in Fig. 8 (top). Clearly, the median size and variance of the class diagrams reported in GUR are larger than in our sample. While the variation is easily explained by the much larger sample size, the difference in diagram size median is a genuine phenomenon. Since the Gothenburg repository is, to our best knowledge, the largest of its kind currently available, this may document a limit of the ecological validity of our findings. On the other hand, the GUR population has been created by harvesting class diagrams from the web, which may in itself amount to a size bias. All we can assert at this point is that our sample is covered by the GUR population.

We have also elicited the number of diagram flaws they contain (Fig. 8, bottom) according to our working definition

[5] For technical reasons, we could not consider line segments, thus the sizes we report for GUR diagrams may sometimes be smaller than those we have used in our study.

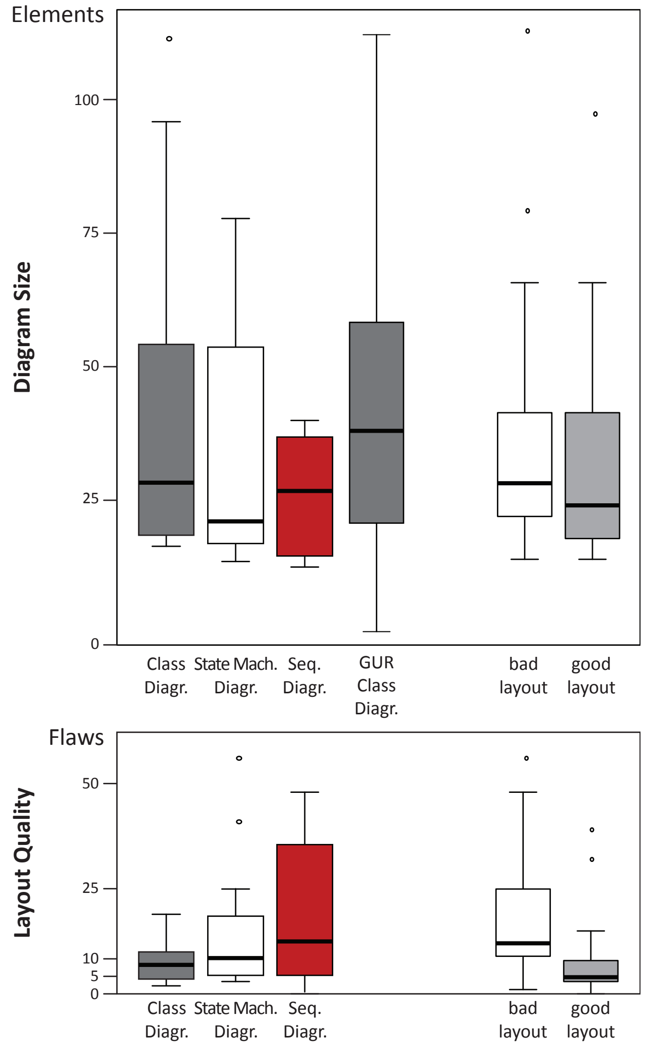


Fig. 8 The sample diagrams analyzed for size and number of flaws, by diagram type and layout quality.

given in Section 2, i.e., the number of line crossings and bends. Clearly, diagrams with bad layouts have substantially more flaws than diagrams with good layouts, precisely that is how we defined "good" and "bad" layouts. It is also clear that the variance varies for diagram types. This is because our definition of size considers textual labels. Thus, the attributes and operations commonly found in Class Diagrams increase the size, but hardly contribute to the number of line crossings and bends.

4.2 Diagram samples and questions

From each of the three case studies we selected one large and one small diagram from each of the three diagram types (Class, State Machine, and Sequence) yielding 18 diagrams. We then trimmed to fit onto a questionnaire page, and created two layout variants for each of these, one adhering to the rules defined in Section 2, and one violating them. In most cases, this amounted to substantial improvement and minor deterioration of the original diagrams for the "good" and "bad" conditions, respectively. This yielded 36 different diagrams. Fig. 1 above shows an example of a pair of good an bad layouts, the complete set can be downloaded from the web[6]. A sample questionnaire can be found online, too.[7]. Fig. 8 (top) shows the comparison by diagram type, Fig. 8 (bottom) shows a summary of the number of diagram flaws (line bends and crossings) by diagram type and layout quality. Fig. 9 shows the sizes of all individual diagrams in the sample, split by type.

A catalog of ten questions was developed for each of the three diagram types. These catalogs have then been adjusted to the other five diagrams of the same type, e.g., we changed the model element names used in the diagrams, the expected answer to questions, or adjusted to the diagram size. These 18 sets of similar questions were then combined with the 36 diagrams to form 36 different sheets with one diagram and ten questions each. For each of the 18 models, there are two sheets with the same questions on the same model appearing once in a good, and once in a bad layout.

Before the first three experiments, different permutations of five different sheets had been created to validate the questionnaires, estimate the time required, and to explore learning and carry-over effects.

We created nine question sheets, combined them into two different sequences, such that in both sequences there are at most five small or large models, and exactly three models of each of the three types. These sequences were then associated with appropriate diagrams such that both sequences had at most five good or bad layouts each. Then, the complement sequences were created, i.e., those two sequences, that had the bad layout corresponding to the good layout found in the original sequence, and vice versa. This way, no participant of any of our experiments was asked to answer two sheets with different layout of the same model, all participants received nine different out of twelve treatments in varying sequences, and all twelve treatments had roughly the same incidence among all questions.

4.3 Participants and completion rates

The participants for experiments D and E were recruited among students from different computer science classes at the Danish Technical University in Lyngby. The participants for experiment F were recruited among elite graduate students and staff from the University of Augsburg.[8] All participants took part voluntarily with no reward or threat and

[6] See `http://bit.ly/1VA1Oho`

[7] See `http://bit.ly/1S7rZd0`

[8] The names of the experiments correspond to those used in [48]: we keep the numbering to allow easier comparison with previous work. Experiments A, B, and C are published in [47].

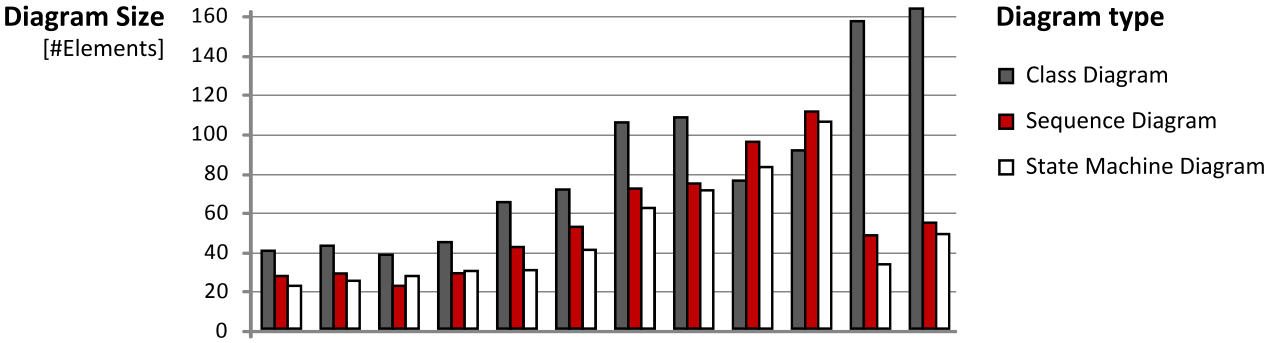


Fig. 9 Distribution of sizes and types of the diagrams used in the experiments: every colored bar represents one diagram of the respective type. Altogether, 36 diagrams were used. Slightly increased bars indicated poorer layouts.

Table 1 Demographic data on the participants of all experiments, "completion" refers to the completion rate on core questions, the aggregated completion in the last row is a weighted average. In order to allow easier cross-referencing, we kept the experiments' identifiers from the original publication [48].

Experiment	Course	Expertise	male	female	all	completion
D	BEng	N	29	3	32	75.1%
E	MSc	I	29	5	34	82.6%
F	Elite	E	10	1	11	90.1%
all	-	-	**68**	**9**	**78**	**80.5%**

under complete anonymity, i.e., it was clear to students that their performance had no influence whatsoever on their grades, for instance. Immediately before the experiment, all participants received a ten-minute introduction to those parts of the UML that were covered in the experiment.

The participants showed a wide spread in UML knowledge. In all experiments, in the core parts of the questionnaire, nine diagrams were presented and ten questions were asked per diagram. We saw an overall completion rate of these core questions of over 80%. See Table 1 for more details on the population.

5 Results

5.1 Correlations between diagram size and modeler performance

As outlined above, our initial hypothesis was that there is a correlation between diagram size and modeler performance in understanding these diagrams. Plotting the diagram size as defined above against the performance on all diagrams yielded the scatter plots shown in Fig. 10. Adding trendlines reveals that the correlation is indeed present: with increasing diagram size, the mean score decreases while the variance increases. Similarly, perceived diagram clarity decreases with increasing diagram size. Surprisingly, there is also a positive correlation between diagram size and perception of layout quality.

We then tested properly for correlations between diagram size and modeler performance. We used the simple diagram size metric, as discussed above, and correlated it with all measures of modeler performance observed in our experiments. We calculated the correlations between diagram size and modeler performance using Pearson's product-moment correlation. We assess the effect size of a correlation of up to 0.3 to as small (S), as large (L) for values over 0.4, and as medium (M) for values in between, see Table 2.

It is quite clear that there is indeed a large correlation between increasing diagram size and decreasing mean scores. This is in line with the observation that the variance increases with diagram size: increased difficulty will provoke a greater spread of results. We have seen a similar effect in our previous studies, where the natural variance in capability of the population becomes more visible when testing poor layouts because these help less with diagram understanding. For the good layouts, individual performance differences matter less, as they are partially leveled by the helpful layout. This objective measure is further confirmed by the subjective measure of asking the participants to assess the clarity of the diagrams: uniformly, large correlations are found between increasing diagram size and decreasing clarity. Yet more confirmation is found when considering the subjective assessment of cognitive load: with increasing size, cognitive load as expressed by subjective assessment of task complexity increases, too. Observe that subjective assessment has been found to be highly correlated with objective measures of cognitive load [12], and that both questions asked to measure cognitive load exhibit similar patterns. The negative correlation between diagram size and perceived di-

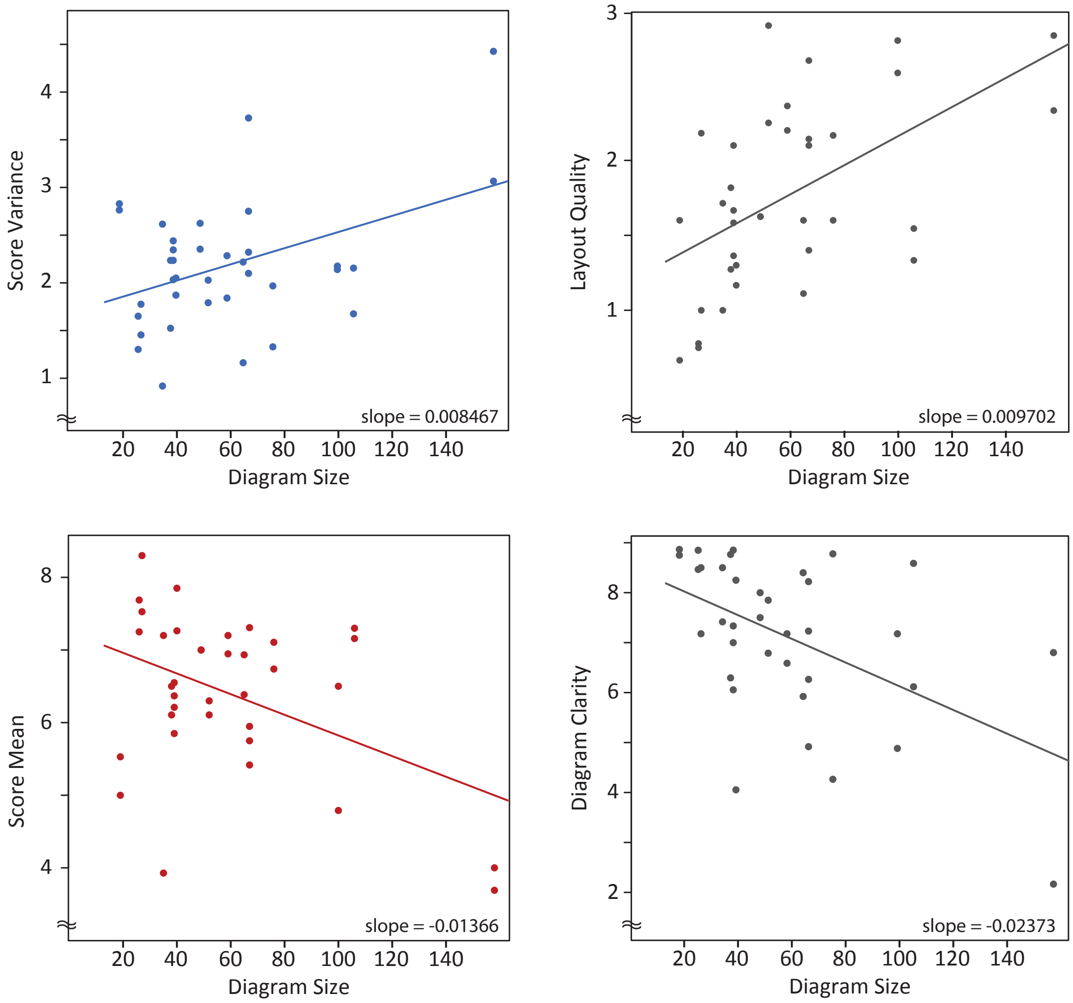


Fig. 10 Plots of various measures of modeler performance against diagram size (clockwise from left bottom): score mean, score variance, subjective assessment of layout quality and diagram clarity. The trend-lines are created from linear models.

agram complexity might be an experimental artifact since it has no statistic significance and relatively small effect sizes.

Confusingly, we also see a positive correlation between diagram size and layout quality (Fig. 10, top right), which seems to contradict our hypothesis. We offer two possible explanations for this phenomenon. The first explanation is that we might see here a weakness of our experimental design, especially since all other findings seem to consistently support our hypothesis. Given the clarity of the finding, however, that seems unlikely: one would expect a weaker trend. Another, more plausible explanation is that participants intuitively understand layout quality *relative to the diagram size*, i.e., quality is not the number of flaws, but the number of flaws relative to the overall size. This would match the usual grading procedure of academic exams. This should be examined more closely in future work.

All of these effects are substantially stronger for poor layouts than for good layouts. This is in support of our initial hypothesis that layout quality matters more with increasing diagram size. In other words: small diagrams are easy to use anyway, so bad layout can be easily compensated. For larger diagrams, however, when the visual and/or mental capacity of a modeler is reached or exceeded, the impact of layout quality becomes visible: layout quality matters more, and is more apparent for larger diagrams.

The results for objective measures and subjective assessments seem to provide stronger results than the results for cognitive load measures. This effect can be seen for all the above measures except the subjective assessments of layout quality and clarity. This is a surprising finding: while it is perfectly intuitive to see a divergence between subjective and objective measures (assessment vs. performance and load), a divergence between the two objective measures is counter-intuitive: one would expect a direct causal connection between load and performance, i.e., decreasing performance with increasing load, and if a factor such as diagram size affects one of them, it should similarly affect the

Table 2 Pearson's product-moment correlation between diagram size and modeler performance, measured as mean and variance of objective performance (correct answers, i.e., score), different subjective assessments, and cognitive load measures. In each cell, the first number is Pearsons' r indicating the size of the correlation, the letter S/M/L classifies the effect size, the next number is the p-value, and the stars indicate its significance level.

Objective Performance	Score Mean				Score Variance			
	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	-0.423	L	0.010	**	0.424	L	0.010	**
Bad Layout	-0.491	L	0.039	*	0.534	L	0.023	*
Good Layout	-0.396	M	0.104	*	0.303	M	0.222	

Question	Answering				Understanding			
	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	0.200	S	0.242		-0.423	L	0.010	*
Bad Layout	0.046	S	0.857		-0.491	L	0.039	*
Good Layout	0.337	M	0.171		-0.396	M	0.104	*

Diagram Assessment	Layout Quality				Layout Clarity			
	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	0.538	L	< 0.001	***	-0.508	L	0.002	**
Bad Layout	0.521	L	0.027	*	-0.563	L	0.015	*
Good Layout	0.573	L	0.013	*	-0.766	L	0.0002	***

Cognitive Load	Diagram Understanding				Diagram Complexity			
	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	-0.338	M	0.044	**	-0.081	S	0.640	
Bad Layout	-0.452	L	0.060	*	-0.313	M	0.207	
Good Layout	-0.197	S	0.434		0.152	S	0.548	

other, and both should show up with more or less the same degree of significance.

Our findings, however, indicate that this is not the case, implying that there is at least one other factor involved. There are two potential candidates for this factor: a methodological flaw or a cognitive process associated to high levels of expertise.

First, our observation could be explained by an experimental artifact showing a weakness in our procedure. For instance, it might be that participants believe that diagram quality is by definition independent or orthogonal to diagram size. Since this is exactly our research question, we intended to communicate neither this interpretation, nor the opposite, but we cannot exclude that there are preconceived notions. Given the large number of participants that should reduce experimenter bias, this seems a rather unlikely explanation.

Second, the observation might be explained by a cognitive process found in experts but not in novices. This process would implement a strategy that allows experts to process diagrams more effectively than novices, in particular to better cope with poor layouts. Such a strategy would have been developed or acquired, and then automated in the process of building up expertise, which is, in fact, the very definition of expertise. We find this second explanation much more convincing than the first. We have therefore explored the differences in expertise levels further in the next section.

5.2 Correlations differentiated by expertise level

Previous work by Abraho, Ricca and others [1,36] suggests that the expertise level is important in diagram understanding, and when controlling for expertise levels, more interesting phenomena become visible (see Table 3). In this table, we have used the same arrangement of values in cells as in Table 2, but have split the data between modelers with lower and higher levels of expertise (left and right, respectively). First of all, we establish that there is indeed a performance difference by expertise levels in our sub-populations. Using a one-sided Wilcoxon-test to compare the average score on good layouts for the two sub-populations, we can reject the hypothesis that the sub-populations exhibit the same performance with very high significance ($p = 0.00013$). When comparing the scores, score variances, and the cognitive load measures, participants with high expertise level are much less affected by increasing diagram size than participants with lower expertise levels. This holds irrespective of layout quality, but is even stronger for poor layouts. Some of these findings are not statistically significant, however, since analyzing the sub-populations separately drastically decreases the number of data points. Still, all correlation show the same pattern and tendencies which does add evidence to our earlier observations.

Even with the reduced population size we find significant or highly significant correlations between increasing diagram size and reduced layout clarity, particularly for poor layout where correlation exceeds -0.7 ($p < 10^{-3}$). Again, the effect is larger for poor layouts than for good ones, and again, the same pattern is found in the cognitive load measures ("Understanding" and "Complexity"), though the latter findings are not statistically significant.

5.3 Quality vs. size

An interesting phenomenon emerges when differentiating between different types of questions. For each diagram, we have asked participants to assess diagrams on an absolute scale regarding both "Layout Quality" and "Diagram Clarity". While the latter is clearly a subjective assessment (also by the question instruction), the former is a question that can be interpreted in a much more objective way.[9] For either question, we see a strong negative correlation to size, i.e., both clarity and quality of larger diagrams are rated worse than for smaller diagrams. However, regarding for subjective assessment ("diagram clarity"), this correlation is much larger for good layouts (almost -0.8) than for bad layouts (-0.563), while there is no such difference when asking for an objective difference ("layout quality").

This means, that participants perceived larger diagrams as more difficult to understanding than smaller diagrams. And while they did not have a clear-cut idea of what constitutes good or bad layout (apart from diagram size), good layout apparently helped with diagram understanding. Clearly, we will need a better, more precise understanding of the notion of layout quality in the future to eventually understand this phenomenon.

5.4 Optimal diagram size

Based on our data, we can compute trend-lines of the correlations, as shown in Fig. 10 (bottom right). Computing a linear model yields coefficients of a linear equation ($\mathit{intercept} = 7.21, \mathit{slope} = -0.014$). This allows us to compute the diagram sizes at which the study participants answered a given number of questions about the diagrams correctly.

Under the assumption that the participants of our studies and the diagrams used are representative for the respective overall populations, we should expect that the scores of all modelers on any UML diagram exhibit a similar distribution. So, disregarding factors such as individual capabilities and diagram layout quality, our experiments lead us to expect a median performance of 6.5 correct answers out of 10 questions asked about a diagram. Or, to put it in another way, we should expect that half the modelers provide 65% of correct answers. It seems natural to us to use the median as a guiding indicator of expected performance.

If we use this indicator together with the linear model we have created before, we can derive a pragmatic guideline for diagram sizes, too. The intercept and slope yielded from correlation data form a linear equation:

$$\mathit{modeler\ score} = \mathit{intercept} + \mathit{slope} \times \mathit{diagram\ size}$$

Filling in the values and resolving for *diagram size* results in a value of 52.3. This corresponds to the expected score of the median of the modeler population. Thus we conclude that diagrams with approximately 50 elements should allow modelers with at least average capabilities to perform at least on an average level when understanding UML State Machine, Class, and Sequence diagrams.

We should expect a degree of variation around this signpost value. Given that the score distribution is skewed towards the maximum (which is an inevitable feature of similar scales), this should be reflected in the size recommendation. We therefore recommend that, in the absence of any more specific information, diagrams should contain in the range of 30 to 60 elements. A geometric interpretation of the relationship between quartiles of score and optimal size is given in Fig. 11. An example of a diagram close to this level is shown in Fig. 12. Section 7 shows, how such a function can be implemented and integrated in a modeling tool to provide guidance to modelers.

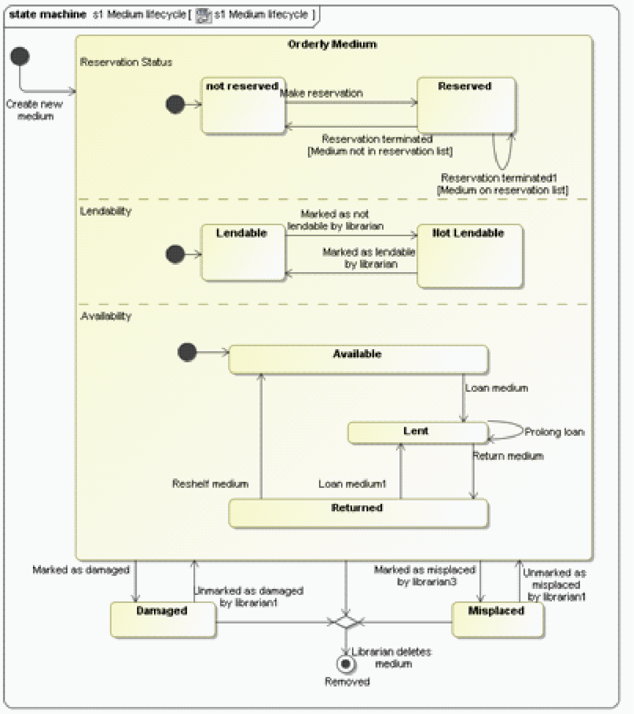

state machine s1 Medium lifecycle [s1 Medium lifecycle]
Orderly Medium
Create new medium
Reservation Status
not reserved
Make reservation
Reserved
Reservation terminated [Medium not in reservation list]
Reservation terminated1 [Medium on reservation list]
Lendability
Lendable
Marked as not lendable by librarian
Not Lendable
Marked as lendable by librarian
Availability
Available
Loan medium
Lent
Prolong loan
Return medium
Reshelf medium
Loan medium1
Returned
Marked as damaged
Unmarked as damaged by librarian1
Damaged
Marked as misplaced by librarian3
Unmarked as misplaced by librarian1
Misplaced
Librarian deletes medium
Removed

Fig. 12 Another sample diagram form the experiments: this diagram contains 63 elements, which is close to the upper bound of size we recommend.

[9] Recall that our participants are students in computer science programs, and are familiar with quality-related concepts like "code smells" or standards like ISO 9126.

Table 3 Pearson's product-moment correlation between diagram size and modeler performance, controlled for expertise level. The cell content has the same arrangement and meaning as in Table 2.

Objective	**Score Mean (low/high expertise)**							
Performance	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	−0.494	L	0.002	**	0.018	S	0.917	
Bad Layout	−0.397	M	0.103	.	−0.173	S	0.493	
Good Layout	−0.615	L	0.007	**	0.243	M	0.331	

Objective	**Score Variance (low/high expertise)**							
Score	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	0.290	M	0.086	.	0.053	S	0.764	
Bad Layout	0.254	M	0.309		0.204	M	0.432	
Good Layout	0.343	M	0.163		−0.085	S	0.736	

Question	**Answering (low/high expertise)**							
	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	0.018	S	0.917		0.274	M	0.105	
Bad Layout	0.409	L	0.092	.	0.282	M	0.257	
Good Layout	−0.313	M	0.206		0.357	M	0.146	

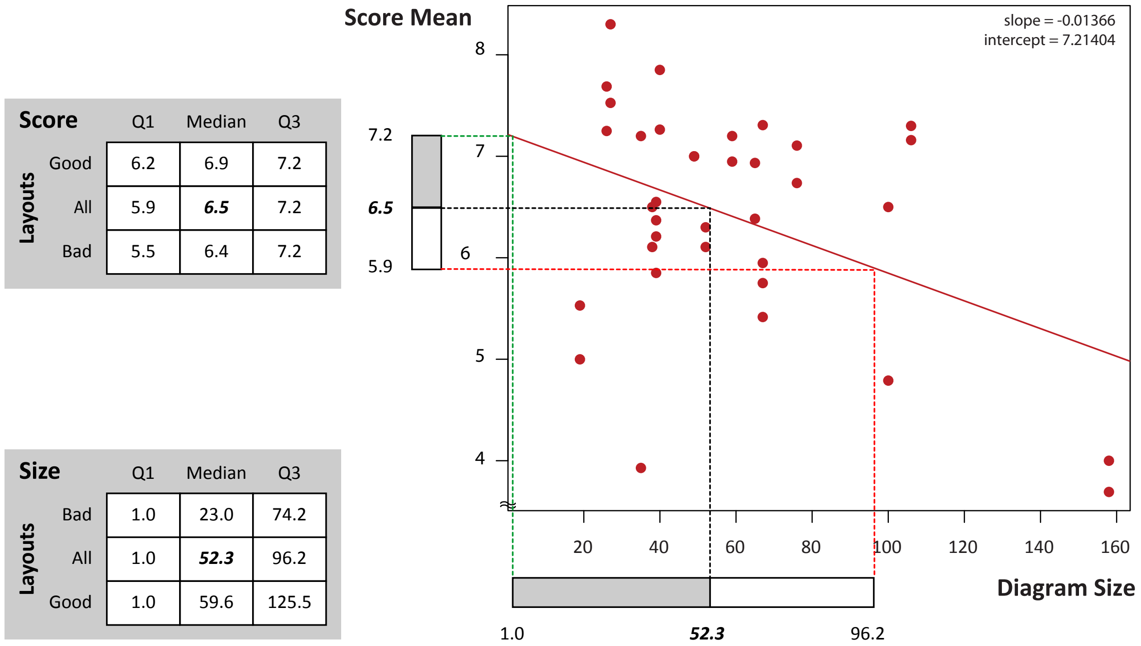
Question	**Understanding (low/high expertise)**							
	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	−0.494	L	0.002	**	0.018	S	0.917	
Bad Layout	−0.397	M	0.103		−0.173	S	0.493	
Good Layout	−0.615	L	0.007	**	0.243	M	0.331	

Diagram	**Layout Quality (low/high expertise)**							
Assessment	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	0.569	L	0.0003	***	0.484	L	0.003	**
Bad Layout	0.534	L	0.023	*	0.516	L	0.028	*
Good Layout	0.615	L	0.007	**	0.536	L	0.022	*

Diagram	**Layout Clarity (low/high expertise)**							
Assessment	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	−0.525	L	0.001	***	−0.440	L	0.007	**
Bad Layout	−0.742	L	0.0004	***	−0.698	L	0.001	**
Good Layout	−0.554	L	0.017	*	−0.570	L	0.014	*

Cognitive	**Diagram Understanding (low/high expertise)**							
Load	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	−0.313	M	0.063	.	−0.199	S	0.245	
Bad Layout	−0.184	S	0.465		−0.064	S	0.800	
Good Layout	−0.421	L	0.082	.	−0.306	M	0.218	

Cognitive	**Diagram Complexity (low/high expertise)**							
Load	r	ES	p	SIG	r	ES	p	SIG
All Diagrams	−0.082	S	0.634		0.042	S	0.808	
Bad Layout	0.133	S	0.600		0.251	M	0.315	
Good Layout	−0.349	M	0.156		−0.134	S	0.595	



Score		Q1	Median	Q3
Layouts	Good	6.2	6.9	7.2
	All	5.9	***6.5***	7.2
	Bad	5.5	6.4	7.2

Size		Q1	Median	Q3
Layouts	Bad	1.0	23.0	74.2
	All	1.0	***52.3***	96.2
	Good	1.0	59.6	125.5

Fig. 11 The red trend-line visualizes the correlation between scores and diagram sizes. The great average of all scores (ignoring individual aptitude and layout quality) is 6.5, which is achieved on diagrams of a size of 52.3 elements. Similarly, the score variance maps into size variance. Geometrically speaking, this means to mirror the distribution of scores at the `size score` trend-line. Observe that high scores correlate to small diagram sizes.

It is important to highlight that this finding dos not imply that creating smaller or larger diagrams should absolutely be avoided. In fact, there are many situations where it is advisable to create much larger diagrams or smaller diagrams. For instance, highlighting a particularly important fact may well warrant a very small diagram with only a handful of elements in it, or the need for a complete overview may lead to very large diagrams with hundreds of elements in them. In the former case, this may just be a poorly utilized area in a document. In the latter, it might become necessary to superimpose higher level structures, or use very large media, such as posters. Both cases occur in practice, and can be cost-effective, when used prudently.

6 Threats to validity

6.1 Internal validity

Great care has been taken to provide systematic permutations of diagrams, questions, and sequences thereof to avoid bias by carry-over effects (“learning”). Any such effects would occur similarly for all treatments and, thus, would cancel each other out. Participants have been assigned to tasks randomly. We can also safely exclude bias through the experimenter himself, since there were only written instructions that apply to all conditions identically. We reduced the danger of introducing bias through the experimental procedure by using several alternative measurements for each variable, thus also corroborating our observations and reducing the implications of poor readings, outliers, and noise.

6.2 External validity

The selection of the models and diagrams may be a source of bias. However, we applied objective and rational criteria to the selection, and compared to previous similar studies, we used three different diagram types (rather than just one or two), a competitively large number of models, and very realistic models. The layouts for the models were, to a large degree, used-as-found, that is, they were created under realistic conditions by people unconnected to these experiments. On top of that, our study is based on a comparatively large number of participants. So, the present study is certainly among the best validated among studies of its kind and we expect our results to be valid for UML models *in general*, i.e., we expect a markedly higher degree of external validity than previous contributions can claim.

6.3 Conclusion validity

We have used non-parametric tests, where applicable, to compensate for skewed distributions in our data. We have consistently provided statistical significance level and the effect size with our inferences. Due to the (relatively) high number of study participants, most of the inferences we present are equipped with high or very high levels of statistic significance and large effect sizes, using Cohn's thresholds for the effect size levels for want of any better guideline. When controlling for sub-populations, the significance levels decrease, but keep showing the same patterns which is sufficient for the claims we make based on these data. We do assume a linear correlation between variables prima facie, but this is justified by an earlier ANOVA-analysis where the squared terms were much too small to have a significant impact on our study.

6.4 Construct validity

Gopher and Braune [12] show that subjective assessment of cognitive load is accurate in the sense that it correlates strongly with objective measures such as skin conductivity, blood oxygenation, pupillary response, or heart rate. Categorizing layout quality as good and bad was done based on existing findings on layout understanding and aesthetics (see Section 8 for more details), which in turn are grounded in the well-established findings of Gestalt psychology.

There is no established metric for "diagram size" in the context of UML or similar notations to which we could liken our own metrics. Therefore, we have developed three different plausible metrics of increasing complexity, but found that they all correlate highly. Thus, we have opportunistically adopted the simplest of these metrics. There is no particular evaluation as to whether this construct is valid.

7 Implementation and Validation

In this section we describe how we have transformed the empirical results obtained above into practical guidelines and a dedicated metrics tool, and how these have been used in the classroom in a modeling course.

We have implemented the DIAGRAMMETRICS tool as part of the MAGICWAND toolset[10] that integrates with the MAGICDRAW UML[11] CASE tool, one of the leading commercial tools for UML-based modeling. It implements the **NODE** metric proposed in Section 3.1 and visualizes the count results in a table. It also aggregates the counts in a gauge and uses color coding to give a recommendation (see Fig. 13). DIAGRAMMETRICS also calculates the total and average counts of sets of diagrams contained in a selection, and displays them simultaneously. The selection can be any subset of diagrams, or a package in the containment tree in which case it counts the diagrams that are contained (recursively) in the tree below the selected package. Such an implementation is easily replicated in other modeling environments.

In order to assess the usefulness of our guideline and tool, we used them in a modeling class for undergraduate students. Then we collected the students' experience as part of the general course feedback sessions. This feedback suggests that the playful visualization is slightly distracting at first, but was generally appreciated by students.

In fact, the visualization seems to be a major attraction. In another modeling-related course students used MAGICDRAW UML, but were not supposed to be using MAGICWAND. During a live demo session, DIAGRAMMETRICS was accidentally activated. The students were very keen on adding MAGICWAND to their toolbox, even for the price of an additional course assignment: a survey of their usage habits. The survey showed that most of the students used the visual feedback permanently, and found it very helpful. One notable quote was *"it's not much of a feedback, but it's always there to remind you not to go overboard"*. A more thorough empirical investigation of the effect of MAGICWAND and DIAGRAMMETRICS in particular is ongoing work.

8 Related Work

The layout of graphs (in the mathematical sense) has been a longstanding research challenge, both with respect to automatic layout and to various aspects of usability, e.g., diagram comprehension, user preferences, and diagrammatic inference. Based on the rich knowledge on general graphs, research on the layout of UML has started with those of UML's notations that are closest to graphs, namely, class diagrams (cf. [38,7,10,55,32]), and, to a lesser extent, communication diagrams (see e.g. [31,34] who use UML 1 terminology). Other types of UML diagrams, in contrast, have only attracted little interest so far (e.g. use case diagrams [8], or sequence diagrams, cf. [2,54]). There is only little work on the Business Process Model and Notation (see [5]), Event Process Chains [20,35], and even less on UML activity diagrams. Most of these report experiments with very small numbers of participants, with the exception of [20] which reports a large scale experiment (n=73) with (brief) follow-up interviews (n=12).

Research on aspects of UML class diagrams has mostly focused on the impact of isolated low-level layout criteria such as line bends, crossings, and length. Unsurprisingly, each of these properties has little impact by themselves and are hard to prioritize. The more elusive higher levels like layout patterns, diagram flow, and the correspondence between a diagram and its intended message seem to have not yet been studied empirically at all. The influence of the expertise level, on the other hand, has been studied [1,36].

The main focus of previous work on UML diagram types and their layout has been with one of four aspects: diagram comprehension (cf. [42,43,29,34] and/or user preference (cf. [32,52]), automatic layout (cf. [7,10,30,8,4]), or one of a

[10] See http://bit.ly/20SXwlk

[11] See www.magicdraw.com

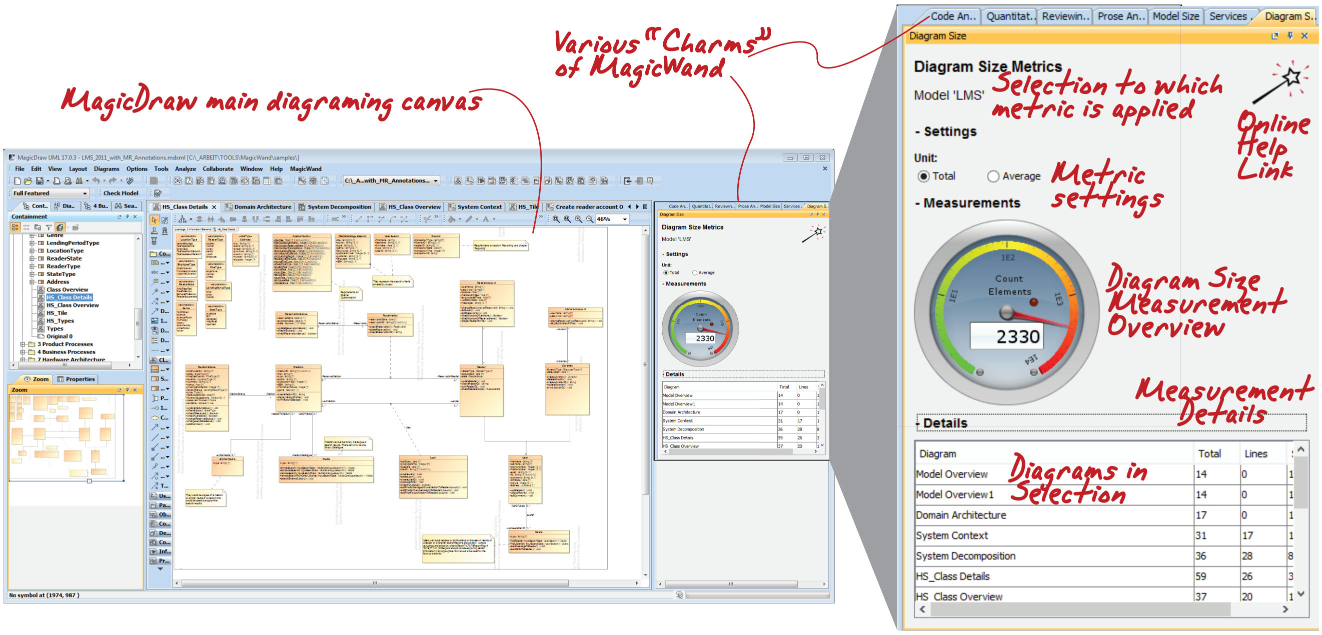


Diagram Size Metrics

Model 'LMS'

- Settings

Unit:

● Total ○ Average

- Measurements

2330

- Details

Diagram	Total	Lines	
Model Overview	14	0	1
Model Overview1	14	0	1
Domain Architecture	17	0	1
System Context	31	17	1
System Decomposition	36	28	8
HS_Class Details	59	26	3
HS_Class Overview	37	20	1

Fig. 13 Screenshot of the MagicWand "Diagram Size"-charm.

variety of diagram inference tasks, e.g., program understanding based on visualizations (cf. [54]), or the role of design patterns in understanding (cf. [43,44]).

Most research uses controlled experiments and evaluate user performance using paper questionnaires, or online surveys. Only a few contributions have used other methods, most notably eye tracking (see [3,55,43]). After using both methods for essentially the same experiment, Sharif et al. have concluded that these two methods are mostly complementary wrt. comprehension tasks (cf. [41]). Thus, eye tracking is only favorable for a tightly restricted set of research questions, in particular when taking into account the considerable cost and effort involved. Having said that, most questionnaire-based approaches employ only very few participants in their experiments, typically in the range of 15 to 30, with the notable exceptions of [42], [33] and [2] involving 45, 55 and 78 participants, respectively. The research done for the current paper involved 78 participants.

More recently, there has been the study of Soh et al. [46], who use eye tracking in a large empirical study (n=21), though they restrict themselves to class diagrams, and are interested in modeler performance only with a view to its predictive value for success in a future professional career. Sharif et al. [40] study 3 different layout schemes of UML class diagrams using both questionnaires and eye tracking on maintenance tasks.

There is a body of theory on visual languages (see [19] for an overview and [13,22] for more recent contributions). All of these are concerned with the language level, that is how to specify or analyze visual languages. For instance, *"The goal of [PoN] is to establish the foundations for a science of visual notation design."* [22, p. 758]. Our work, in contrast, focuses on the *statements* expressed using a visual language such as UML, that is, concrete diagrams. While existing visual language theory focuses on issues of syntax and semantics, our work focuses on issues on pragmatics, bordering on syntax.

Diagram layout has been classified as "secondary notation", suggesting it somewhat less important than the "proper" elements of a language (i.e., the graphemes, see [11]) and the rules governing their composition into diagrams (i.e., a language's grammar, see [37]). However, Oberlander pointed out that "*many possible layouts are pragmatically inappropriate*" (cf. [23, p. 8]) and suggested it would be highly desirable to avoid "*unwanted graphical implicatures*" (ibid), and Moody concurs "*visual noise (accidental secondary notation) [...] conflicts with or distorts the intended message*" (cf. [22, p. 760]). Furthermore, Petre [26] found that effective use of secondary notation was a significant contributory factor to the effectiveness of a diagram, and the major distinguishing feature between expert and novice use of a notation.

9 Conclusions

9.1 Summary

In earlier work, we established that layout quality does impact the understanding of UML diagrams [47], and that this applies irrespective of diagram type, but dependent on modeler expertise [48]. We could so far not answer the question whether diagram size had an influence, and, if so, what its magnitude would be. Thus, in this paper, we developed metrics for the size of UML diagrams. Since these metrics correlate almost perfectly on a population of 38 diagrams, we concluded that it is irrelevant which of these diagram size metrics is used. Thus we chose the pragmatically simplest metric. Our results suggest that the number of diagram elements is a useful metric for diagram size. Our experimental result suggest preferred levels for this metric.

9.2 Findings

Using this diagram size metric, we re-analyzed existing data sets and find strong evidence in support of our hypothesis. We conclude that high layout quality is particularly helpful for large diagrams, and that it is particularly helpful for modelers with low expertise. Based on these findings, we derive a pragmatic guideline on the optimal size of diagrams Section 5.4 that is (1) very easy to apply in tools, (2) based on objective findings, and (3) promise to be beneficial to many modelers.

9.3 Relevance

The findings reported in this article, as well as the metrics and guidelines proposed may appear straightforward, even obvious to some. However, we maintain that it is not just worthwhile to provide evidence also for the seemingly obvious, but in fact indispensable to seek this evidence. Also, in the experience of the author, there are actually many people that do not find layout quality obviously relevant. Recall that the diagrams used in our study have been provided by students as part of their course assignment, including those diagrams rated poorly by the study participants. So, the modelers could not or would not create better diagrams.

9.4 Validity

The experimental procedure has been designed carefully to exclude bias of any kind, learning effects, and distortion. We have included a relatively large number of participants ($n = 78$) in our experiments, as a further contribution to validity. Most of the tests and correlations we have computed are equipped with high or very high levels of statistical significance. We consistently report completion rates, effect sizes, and similar data to allow scrutinizing our results, and allow other scientists to conduct secondary research based on our work. Thus we conclude, that our findings have a high level of validity.

9.5 Future Work

Consistent with previous findings reported in [47,48,49], a stronger effect is seen in subjective measures (cognitive load, assessment) than in objective measures (score), pointing to cognitive mechanisms to cope with diagram complexity. We hypothesize that increasing extrinsic cognitive load will lead to stronger effects in the objective measures. One way of doing this is through dual-stimulus experiments [28, p. 264].[12] Another avenue of study is the usage of other methods to explore the inner workings of the human brain, such as eye tracking, or advanced imaging techniques (such as functional Magnetic Resonance Tomography, see [45]). In [50, 18] we have reported the results of a pilot study using eye tracking and a sub-sample of the stimuli used in the study underlying the present paper. We hope this constitutes the first steps towards a theory of diagram understanding.

References

1. Abrahão, S., Gravino, C., Insfrán, E., Scanniello, G., Tortora, G.: Assessing the Effectiveness of Sequence Diagrams in the Comprehension of Functional Requirements: Results from a Family of Five Experiments. IEEE Txn. SE **39**(3), 327–342 (2013)
2. Britton, C., Kutar, M., Anthony, S., Barker, T., Beecham, S., Wilkinson, V.: An empirical study of user preference and performance with UML diagrams. In: Proc. IEEE 2002 Symp. Human Centric Computing Languages and Environments (HCC/LE), pp. 31–33. IEEE (2002)
3. Dawoodi, S.Y.: Assessing the Comprehension of UML Class Diagrams via Eye Tracking. Master's thesis, Kent State University (2007)
4. Dwyer, T., Lee, B., Fisher, D., Quinn, K.I., Isenberg, P., Robertson, G., North, C.: A Comparison of User-Generated and Automatic Graph Layouts. IEEE Txn. Visualization and Computer Graphics **15**(6), 961–968 (2009)
5. Effinger, P., Jogsch, N., Seiz, S.: On a Study of Layout Aesthetics for Business Process Models Using BPMN. In: Proc. 2nd Intl. Ws. Business Process Modeling Notation (BPMN), pp. 31–45. Springer Verlag (2010)
6. Eichelberger, H.: Aesthetics of class diagrams. In: Proc. 1st Intl. Ws. Visualizing Software for Understanding and Analysis (VISSOFT), pp. 23–31. IEEE (2002)
7. Eichelberger, H.: Aesthetics and automatic layout of UML class diagrams. Ph.D. thesis, University of Würzburg (2005)
8. Eichelberger, H.: Automatic layout of UML use case diagrams. In: Proc. 4th ACM Symp. Sw. Visualization (SOFTVIS), pp. 105–114. ACM (2008)
9. Eichelberger, H., Schmid, K.: Guidelines on the aesthetic quality of UML class diagrams. Information and Software Technology **51**(12), 1686–1698 (2009)
10. Eiglsperger, M.: Automatic layout of UML class diagrams: a topology-shape-metrics approach. Ph.D. thesis, Univ. Tübingen (2003)

[12] In dual stimulus experiments (also "dual task design"), participants have to do a second, unrelated task concurrently to add enough additional mental load on participants to exceed their capacity. This results in closer correspondence between objective score and subjective assessment of difficulty.

11. Fish, A., Störrle, H.: Visual qualities of the Unified Modeling Language: Deficiencies and Improvements. In: P. Cox, J. Hosking (eds.) Proc. IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), pp. 41–49. IEEE Computer Society (2007)
12. Gopher, D., Braune, R.: On the Psychophysics of Workload: Why Bother with Subjective Measures? Human Factors **26**(5), 519–532 (1984)
13. Green, T.R.G., Petre, M.: Usability analysis of visual programming environments: A 'cognitive dimensions' framework. J. Visual Languages and Computing (7), 131–174 (1996)
14. Gurr, C.A.: Effective Diagrammatic Communication: Syntactic, Semantic and Pragmatic Issues. J. Visual Languages and Computing **10**, 317–342 (1999)
15. Karasneh, B., Chaudron, M.: Online Img2UML Repository: An Online Repository for UML Models. In: Intl. Ws. Experiences and Empirical Studies in Software Modelling (EESSMod) (2013). Co-located MoDELS'13. Repository available online at `http://cse-poros.cse.chalmers.se`
16. Koffka, K.: Principles of Gestalt Psychology. Routledge & Kegan Paul (1935)
17. Köhler, W.: Die physischen Gestalten in Ruhe und im stationären Zustand. Verlage der philosophischen Akademie (1924)
18. Maier, A.M., Baltsen, N., Christoffersen, H., Störrle, H.: Towards Diagram Understanding: A Pilot-Study Measuring Cognitive Workload Through Eye-Tracking. In: Proc. Intl. Conf. Human Behavior in Design (2014)
19. Marriott, K., Meyer, B. (eds.): Visual Language Theory. Springer Verlag (1998)
20. Mendling, J., Reijers, H.A., Cardoso, J.: What Makes Process Models Understandable? In: G. Alonso, others (eds.) Proc. Intl. Conf. Business Process Management (BPM), pp. 48–63. Springer Verlag (2007)
21. Miller, G.A.: The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. The Psychological Review **63**, 81–97 (1956)
22. Moody, D.L.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Trans. Software Engineering **35**(6), 756–779 (2009)
23. Oberlander, J.: Grice for Graphics: Pragmatic Implicature in Network Diagrams. Information Design Journal **8**(2), 163–179 (1996)
24. OMG: OMG Unified Modeling Language (OMG UML). Version 2.5. Tech. rep., Object Management Group (2013). ptc/2013-09-05
25. Paas, F., Tuovinen, J.E., Tabbers, H., Van Gerven, P.W.: Cognitive Load Measurement as a Means to Advance Cognitive Load Theory. Educational Psychologist **38**(1), 63–71 (2003)
26. Petre, M.: Why Looking Isn't Always Seeing: Readership Skills and Graphical Programming. Comm. ACM **38**, 33–44 (1995)
27. Pfleeger, S.L.: Experimental design and analysis in software engineering. Annals of Software Engineering **1**(1), 219–253 (1995)
28. Plass, J.L., Moreno, R., Brünken, R.: Cognitive Load Theory. Cambridge Univ. Press (2010)
29. Purchase, H., Colpoys, L., Carrington, D., McGill, M.: UML Class Diagrams: An Emprical Study of Comprehension. In: K. Zhang (ed.) Software-Visualization: From Theory to Practice, pp. 149–178. Kluwer (2003)
30. Purchase, H.C.: Metrics for Graph Drawing Aesthtetics. J. Visual Languages and Computing **13**(5), 501–516 (2002)
31. Purchase, H.C., Allder, J.A., Carrington, D.A.: Graph layout aesthetics in UML diagrams: user preferences. J. Graph Algorithms Applications **6**(3), 255–279 (2002)
32. Purchase, H.C., Carrington, D., Allder, J.A.: Empirical Evaluation of Aesthetics-based Graph Layout. J. Empirical Software Engineering **7**(3), 233–255 (2002)
33. Purchase, H.C., Carrington, D.A., Allder, J.A.: Experimenting with aesthetics-based graph layout. In: M. Anderson, P. Cheng, V. Haarslev (eds.) Proc. Intl. Conf. Theory and Application of Diagrams (Diagrams), no. 1889 in LNAI, pp. 489–501. Springer Verlag (2000)
34. Purchase, H.C., Colpoys, L., McGill, M., Carrington, D.: UML Collaboration Diagram Syntax: An Empirical Study of Comprehension. In: Proc. 1st Intl. Ws. Visualizing Software for Understanding and Analysis (VISSOFT), pp. 13–22. IEEE CS (2002)
35. Recker, J., Dreiling, A.: Does It Matter Which Process Modelling Language We Teach or Use? An Experimental Study on Understanding Process Modelling Languages without Formal Education. In: M. Toleman, A. Cater-Steel, D. Roberts (eds.) Proc. 18^{th} Australasian Conf. Information Systems. University of Southern Queensland (2007). URL `http://eprints.qut.edu.au/12270`
36. Ricca, F., Di Penta, M., Torchiano, M., Tonella, P., Ceccato, M.: How Developers' Experience and Ability Influence Web Application Comprehension Tasks Supported by UML Stereotypes: A Series of Four Experiments. IEEE Txn. SE **36**(1), 96–118 (2010)
37. Schürr, A., Klar, F.: 15 Years of Triple Graph Grammars. Research Challenges, New Contributions, Open Problems. In: H. Ehrig, et al. (eds.) Intl. Conf. Graph Transformation (ICGT'08), no. 5214 in LNCS, pp. 411–425. Springer Verlag (2008)
38. Seemann, J.: Extending the Sugiyama algorithm for drawing UML class diagrams: Towards automatic layout of object-oriented software diagrams. In: Proc. Intl. Conf. Graph Drawing (GD), pp. 415–424. Springer (1997)
39. Selic, B., Kent, S., Evans, A. (eds.): Proc. 3^{rd} Intl. Conf. Unified Modeling Language (UML'00), no. 1939 in LNCS. Springer Verlag (2000)
40. Sharif, B.: Empirical Assessment of UML Class Diagram Layouts Based on Architectural Importance. In: Proc. Intl. Conf. Software Maintenance (ICSM), pp. 544–549. IEEE (2011)
41. Sharif, B., Maletic, J.I.: An empirical study on the comprehension of stereotyped UML class diagram layouts. In: Proc. 17th IEEE Intl. Conf. Program Comprehension (ICPC), pp. 268–272. IEEE (2009)
42. Sharif, B., Maletic, J.I.: The effect of layout on the comprehension of UML class diagrams: A controlled experiment. In: Proc. 5th Intl. Ws. Visualizing Sw. for Understanding & Analysis (VISSOFT), pp. 11–18. IEEE (2009)
43. Sharif, B., Maletic, J.I.: An eye tracking study on the effects of layout in understanding the role of design patterns. In: Proc. 2010 IEEE Intl. Conf. Software Maintenance (ICSM), pp. 41–48. IEEE (2010)
44. Sharif, B., Maletic, J.I.: The Effects of Layout on Detecting the Role of Design Patterns. In: Proc. 23rd IEEE Conf. Software Engineering Education and Training (CSEE&T), pp. 41–48. IEEE (2010)
45. Siegmund, J., Kästner, C., Apel, S., Parnin, C., Bethmann, A., Leich, T., Saake, G., Brechmann, A.: Understanding Understanding Source Code with Functional Magnetic Resonance Imaging. In: Proc. ACM/IEEE Int. Conf. Software Engineering (ICSE), pp. 378–389. ACM Press (2014)
46. Soh, Z., Sharafi, Z., Van den Plas, B., Porras, G.C., Guéhéneuc, Y.G., Antoniol, G.: Professional Status and Expertise for UML Class Diagram Comprehension: An Empirical Study. In: Proc. Intl. Conf. Program Comprehension (ICPC), pp. 163–172. IEEE (2012)
47. Störrle, H.: On the Impact of Layout Quality to Unterstanding UML Diagrams. In: Proc. IEEE Symp. Visual Lang. and Human-Centric Computing (VL/HCC), pp. 135–142. IEEE Computer Society (2011)
48. Störrle, H.: On the Impact of Layout Quality to Unterstanding UML Diagrams: Diagram Type and Expertise. In: G. Costagliola,

others (eds.) Proc. IEEE Symp. Visual Languages and Human-Centric Computing (VL/HCC), pp. 195–202. IEEE Computer Society (2012)

49. Störrle, H.: On the Impact of Layout Quality to Understanding UML Diagrams: Size Matters. In: J. Dingel, others (eds.) Proc. 17th Intl. Conf. Model Driven Engineering Languages and Systems (MoDELS), no. 8767 in LNCS, pp. 518–534. Springer Verlag (2014)
50. Störrle, H., Baltsen, N., Christoffersen, H., Maier, A.M.: On the Impact of Diagram Layout: How Are Models Actually Read? In: S. Sauer, others (eds.) Joint Proc. MODELS 2014 Poster Session and ACM Student Research Competition, vol. 1258, pp. 31–35. CEUR (2014)
51. Störrle, H., Fish, A.: Towards an Operationalization of the "Physics of Notations" for the Analysis of Visual Languages. In: A. Moreira, B. Schätz, J. Gray, A. Vallecillo, P. Clarke (eds.) 16th Intl. Conf. Model Driven Engineering Languages and Systems (MoDELS'13), no. 8107 in LNCS, pp. 104–120. Springer Verlag (2013)
52. Swan, J., Kutar, M., Barker, T., Britton, C.: User Preference and Performance with UML Interaction Diagrams. In: Proc. 2004 IEEE Symp. Visual Languages and Human Centric Computing (VL/HCC), pp. 243–250. IEEE (2004)
53. Tomonaga, M., Matsuzawa, T.: Perception of Complex Geometric Figures in Chimpanzees (Pan troglodytes) and Humans (Homo sapiens): Analyses of Visual Similarity on the Basis of Choice Reaction Time. Journal of Comparative Psychology **106**(1), 43–52 (1992)
54. Wong, K., Sun, D.: On evaluating the layout of UML diagrams for program comprehension. Software Quality J. **14**(3), 233–259 (2006)
55. Yusuf, S., Kagdi, H., Maletic, J.I.: Assessing the Comprehension of UML Class Diagrams via Eye Tracking. In: 15th IEEE Intl. Conf. Program Comprehension (ICPC'07), pp. 113–122. IEEE CS (2007)
56. Zimbardo, P.G.: Psychology, 18th intl. edn. Pearson Education (2007)