Insights for Improving Diagram Editing Gained from an Empirical Study

Michael Stegmaier
Institute for Software Engineering and Programming Languages
Ulm University
Ulm, Germany
michael-1.stegmaier@uni-ulm.de

Alexander Raschke
Institute for Software Engineering and Programming Languages
Ulm University
Ulm, Germany
alexander.raschke@uni-ulm.de

Matthias Tichy
Institute for Software Engineering and Programming Languages
Ulm University
Ulm, Germany
matthias.tichy@uni-ulm.de

Eva-Maria Meßner
Dept. Clinical Psychology and Psychotherapy
Ulm University
Ulm, Germany
eva-maria.messner@uni-ulm.de

Sara Hajian
Ulm University
Ulm, Germany
sara.hajian@uni-ulm.de

Anna Feldengut
Ulm University
Ulm, Germany
anna.feldengut@uni-ulm.de

***Abstract*—The modeling process plays an important role in software and systems engineering. Models are often created on paper or whiteboards even though software for modeling exists. We conducted an empirical study to identify how diagram editing can be improved by investigating how people model on whiteboards. The study was designed as a qualitative study and had 34 participants from four different education fields. The participants were recorded in video and audio while performing two tasks in pairs on a whiteboard. In the first task they had to model a subset of a network plan of the local public transport system. In the second task they had to model the daily schedule of a canteen kitchen. After finishing the tasks, we conducted a semi-structured interview. Using this study, we identified minor different preferences between different education fields and gained several insights for improving diagram editing, e. g., support for drawing sketches concurrently and complementary to models, supporting flexible mechanisms for highlighting different parts of a model, better support for making shapes in the model look consistent.**

***Keywords*-software engineering; systems engineering; modeling; usability; tools; empirical study; whiteboard; human factors; human computer interaction**

I. Introduction

In software and systems engineering, graphical representations play a key role during the modeling process. Graphical representations help to get an overview of a model or system and facilitate the understanding of the relationships between its components. Over the years, several modeling languages and associated graphical notations have emerged. Prominent ones are Unified Modeling Language (UML) [1] and its derivative Systems Modeling Language (SysML) [2]. Many different tools support these modeling languages and graphical editing using the graphical notations.

We have often heard from software engineers in industry and academia that for modeling they prefer to use tools like Adobe Illustrator or Microsoft PowerPoint rather than the available modeling tools. The reason for this could be the poor usability of modeling tools [3]–[5]. Furthermore, studies (e. g., [6]) confirm our own experience that whenever we discuss a model or plan software using a model we draw it onto a whiteboard instead of using specialized modeling software. This is the case although whiteboards have several known limitations, such as limited space, no undo functionality (especially when content has been erased) and the fixed location in a room [7]. Typically, the drawn model is either kept on the whiteboard during development or a photo of it is taken. In most cases, we do not bother to recreate the model using modeling software. It seems like there is something about modeling software that keeps us from using it while there is something about whiteboards that attracts us [8]. What keeps us from using modeling software most likely is their poor usability. So the question is, what makes modeling on whiteboards with all their limitations more attractive than modeling with modeling software.

To better understand this, we designed and conducted an empirical study on modeling behavior on a whiteboard. The insights gained from this study should help to understand how diagram editing in modeling tools can be improved. The study participants have to perform two tasks on a whiteboard working in pairs. In total, we gathered 34 participants from four different education fields for our study.

With this study we want to identify how people create and interact with models intuitively without being limited to the palette of tools and actions provided by software. In particular we want to answer the following questions:

- RQ1 How do people interact with a whiteboard while modeling on it?
- RQ2 What do people wish from modeling tools compared to the facilities of a whiteboard?

To improve internal validity of our results we not only

let the participants perform the described tasks while being recorded in video and audio but also ask them about their decisions and preferences afterwards in a semi-structured interview. The intention of domain specific languages and modeling software based on them is to enable domain experts to work with models themselves. So to improve external validity, we gathered participants from different educations fields. Due to having participants from various education fields, we cannot use software modeling languages like class diagrams or activity diagrams in our tasks. We have to use something that everyone can work with. So we decided to use a network plan for the local public transport system for the first task and the daily schedule of a canteen kitchen for the second task. The first one covers modeling structure in a graph-like language (similar to component diagrams). The second covers modeling behavior (slightly similar to sequence diagrams).

All recordings were coded and subsequently analyzed. In the end we have gained eight insights which are described in Section IV.

After discussing related work in Section II, we describe the study design in Section III and the results in Section IV. Threats to validity are discussed in Section V and finally, we conclude this paper in Section VI.

II. Related Work

The use of whiteboards has already been investigated in several studies. However, the aspects studied differ considerably:

Damm et al. "conducted two field studies of software developers using CASE tools and whiteboards" [9]. They observed the participants and then interviewed them focusing on the following three aspects of design activities: *Cooperation* including communication, coordination, and collaboration aspects of software design. *Action* describes the physical interaction with the tools resp. board and *use* involves the semantics of the resulting diagrams / sketches. The presented observations are based on informal notations of an observer who attended the design sessions. The reported "actions" are not as detailed as we identified in our study. In contrast, Damm et al. describe, e. g., who was drawing what kind of diagram or who takes a photograph of the results. Therefore, the concluded criteria for a modeling tool are rather high-level like "Provide a direct and fluid interaction", "Integrate formal, informal, and incomplete elements". Although this information is important for a tool vendor, we focus on more detailed actions like in which order are the elements drawn, how often do the users change a pen.

The interaction between the participants of a design session is the topic of a study performed by Rooksby and Ikeya [10]. They focus on how the developers "remain coordinated and focused while collaborating at a whiteboard" by analyzing, e. g., the movement in the room, eye-contacts, and nonverbal communication.

Mangano et al. [11] published a literature study on how designers draw, how they navigate drawings and how they collaborate with drawings. They identified 14 behaviors that occur during design at the whiteboard. These behaviors are again high-level and include for example that "designers draw different kinds of diagrams" or they "draw what they need, and no more". The authors derive design recommendations for modeling tools from their empirical results. In our study we focus on a more fine-grained level of interactions.

Another study of Mangano et al. [12] has been conducted with a methodology similar to ours. The pairwise modeling of 18 professional software designers at a whiteboard was recorded. The resulting 14 hours of video material was coded. The difference is that they analyzed the produced content at the whiteboard: What kinds of notations are used? What visual syntactical elements support the design activity? How do designers interact with different drawings? The results are discussed in the context of requirements for new electronic software design tools. In contrast, we do not focus on the content, but on how the participants interact with the whiteboard and pens. Furthermore, our study differs from theirs in that our participants are from different education fields.

Similarly, Petre studies the cognitive dimensions [13] of the representations software designers use on the whiteboard [14], but the applied method was different. The cognitive dimensions framework defines 13 aspects "capturing a significant amount of the psychology and HCI of programming" [13]. These dimensions are a set of terms that are easy to understand and can be retrieved by answering corresponding questions. The difference to our approach is again the focus on the notations and representations instead of the activity itself.

The use of whiteboards is also subject in a study described by Mynatt [15]. For this study, every day for two weeks photos were taken of the whiteboards of the 18 participants. Additionally, a questionnaire and an interview were carried out. Based on the observations, four profiles of the analyzed whiteboard users are identified. The observations included several aspects like managing space, the types of tasks for which the whiteboard has been used, the frequency of use, and how the whiteboard is used in collaboration with other people. Based on these results, Mynatt describes a list of desired whiteboard features especially for electronic versions of it.

Chen et al. report several characteristics of whiteboards' advantages and disadvantages compared to existing modeling tools [16]. However, these were not collected by a study and were only described informally.

In summary, several different aspects regarding the use of whiteboards for modeling activities have been investigated, but none of them focused on the fine grained drawing

Table I
DISTRIBUTION OF PARTICIPANTS

Education	Number of Participants	Distribution Female	Male
Computer Science	8	2	6
Psychology	8	7	1
Design	8	4	4
Mathematics	6	2	4
Other	4	4	0
Total	34	19	15

Table II
MEAN AND STANDARD DERIVATION OF THE RESULTS OF THE TA-EG QUESTIONNAIRE

Education	Mean	Standard Derivation
Computer Science	3.78	0.38
Psychology	3.30	0.58
Design	3.65	0.46
Mathematics	3.23	0.31
Other	3.05	0.40
Total	3.48	0.50

actions of whiteboard users, which might give hints for the improvement of existing modeling tools.

III. STUDY DESIGN

The study that has been conducted is described below. The study is designed to have two participants working together on a whiteboard for one hour. While working on the whiteboard they are recorded by a camera that captures video and audio. The participants have to perform two different tasks. The first task is about modeling the structure of a system while the second task is about modeling the behavior of a system. The difference between these two tasks should reveal different interaction patterns. In order to better understand the interaction patterns, the participants are asked to work together and to discuss their decisions. A pair of participants has to have the same or very similar education so we can draw conclusions from their education. The study was always conducted in the same room with the same setup, i.e., the same camera pointing at the whiteboard in roughly the same angle, the same markers, eraser and magnets.

A. Participants

We split the participants into five different categories depending on their education. Table I shows the distribution of our participants in categories. Unfortunately, we could not get one more pair in Mathematics to have 8 participants for each of the main categories. The gender distribution is almost balanced with 19 female participants compared to 15 male participants. The category "Other" contains students studying Medicine and secretaries. Table II shows the results of the TA-EG questionnaire. This questionnaire measures the participants' affinity to technology. The values range from 1 to 5 with 5 being the highest affinity and 1 being the lowest. It shows that no participant had significantly low affinity to technology.

B. Questionnaires

Before the participants can start working on their tasks, they are asked to answer two questionnaires. The first one records demographic data such as age and education while the second one records technical affinity. The second questionnaire is the TA-EG questionnaire [17] (original German title: Technikaffinität – Elektronische Geräte, translated: Technology Affinity – Electronic Devices).

C. Task 1

In the first task, the participants have to model a subset of a network plan of the local public transport system. For this, the participants get a description of the network plan written in natural language. To perform this task, the participants are provided with whiteboard markers with four different colors (red, green, blue and black), an eraser and three magnets. The description contains a tram, a bus and a train. For tram and bus it provides a sequence of bus stops. For train it lists the stations with their connections and how long each connection takes. When the participants are done with this task they get a description of how they have to change their plan. This change involves replacing stations served by the bus and adding a new tram to the plan. So if the participants used one color for bus stops and stations and one color for each line they will run out of colors for the new tram now. Figure 1 shows one of the solutions for this task. In this example we can see that the participants used a dashed line when they ran out of colors for the second tram.

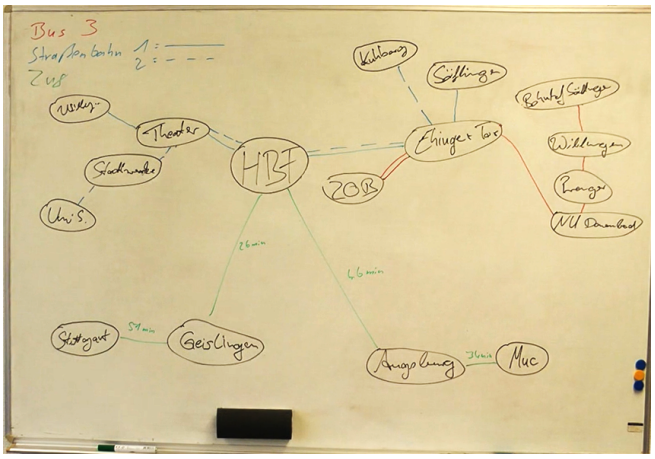

Figure 1. Example solution for Task 1

D. Task 2

In the second task, the participants have to model the daily schedule of a canteen kitchen. For this, the participants get a description of the schedule written in natural language. The schedule describes tasks and who is responsible for each task. For each task it is mentioned when it should start and

when it should be finished or how long it takes. Additionally, it describes dependencies between different tasks that also have to be visible in the resulting model. Analogous to the first task, the participants are provided with the same four whiteboard markers, an eraser and three magnets. When the participants are done with this task they get a description of how they have to change their plan. This change involves splitting up one task into two tasks, adding a new task and changing the duration of one task. Figure 2 shows an example solution for this task. In this example we see some kind of sequence diagram with time.

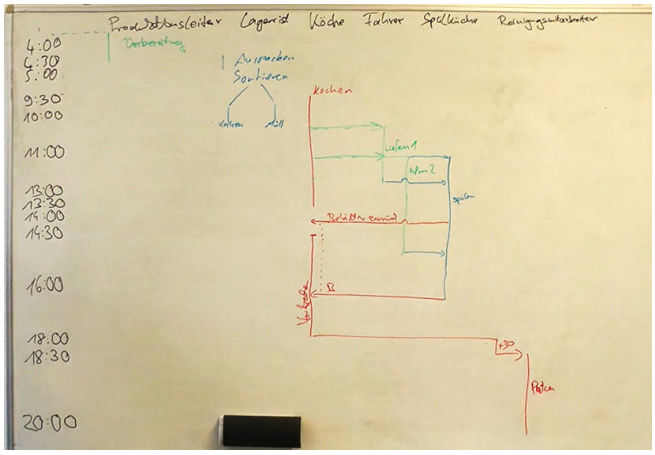

Figure 2. Example solution for Task 2

E. Interview

Immediately after each of the two tasks, a small, semi-structured interview with the participants begins. This interview starts with the question if the participants are confident with their solution or if they wanted to improve something. After that, they are asked questions about the decisions they made in their model. Now that they have been confronted with questions about their decisions, they are asked if they would publish their model as it currently is. Finally, the last question is what tools they wished to have in a software if they had to perform the same task using a computer. While the first two questions help us to understand their decisions, they also make the participants take another look at their whole model and draw their attention to their decisions. The third question aims at determining whether the model is just a sketch or a final version of their model. The goal of the last question is to learn what people demand from modeling software.

IV. Results

After illustrating the results of the questionnaires, we discuss the results of the study and interviews in this section.

For the qualitative results, three of us coded the videos. To develop a common coding system, each of us coded the same video independently. While discussing the results we tried to unify the codes we found to build a common set of codes. With this common set we coded another video independently from each other to see how well we agree and how well the codes work so far. In the end, we have 65 different codes in our coding system. A code in a video begins when a marker / eraser starts touching the whiteboard and ends when it stops touching it. These codes can be split into eight categories as shown in Table III which is sorted by number of applications. The eighth category is not listed in this table as it differs from the other categories. It is the category "Interview" which contains the codes that are applied when the participants answer the questions of the interviews. Codes from this category were applied 368 times. It is separated from the other categories because it is not directly related to the modeling. This category contains the most different codes. This is because there is a variety of different suggestions and preferences among the participants. Most participants wished they had a way to move around elements within their model. Others wished an infinite canvas, more colors or tools such as a grid or rulers for alignment.

Table III
Codes and their number of applications

Name	Description	Number of appl.
Creation	**Creation of an element**	**1081**
Labelling	**Labelling of an element**	**830**
Organization	Decisions about colors / shapes	380
Deletion	Deletion of an element	229
Modification	Modification of an element	171
Sketch	**Sketch, independent from model**	**30**
Miscellaneous	Unique codes without category	10
Total	Total number of code applications	2731

Out of 16 hours and 5 minutes of video material there are 3 hours and 39 minutes of interviews and out of the remaining 12 hours and 26 minutes there are 4 hours and 34 minutes of coded video segments. This means that the ratio between coded video segments and uncoded video segments is 36.7% to 63.3%. So the participants spent roughly half of the time drawing / erasing something on the whiteboard as they spent on doing something else such as thinking and discussing.

A. Interviews

After each task, the participants were interviewed. That means we gathered 34 interviews which sum up to 3 hours and 39 minutes of interview material. The wishes mentioned in these interviews are listed and described in Table V. It is not surprising that most participants wished to have a way to move around elements on their canvas as this is one of the major disadvantages of a whiteboard compared to software.

B. Additional Observations

We observed that during the development of the models several participants created quick sketches that they erased

Table IV
DIFFERENCES IN AVERAGE NUMBER OF APPLICATIONS OF CODES PER VIDEO COMPARING THE DIFFERENT EDUCATION FIELDS

Code (number of pairs)	CS (4)	Psych. (4)	Design (4)	Math (3)	Other (2)
Creation / Line	38.25	28.50	43.50	36.00	39.00
Creation / Circular Shape	12.00	14.25	29.25	8.33	7.50
Creation / Rectangle	5.50	5.25	5.25	9.33	7.50
Labelling / Inside / Before	**9.50**	**1.00**	**4.00**	**3.00**	**0.00**
Labelling / Inside / After	**3.00**	**3.25**	**9.25**	**5.67**	**12.00**
Sketch	**0.75**	**2.75**	**1.00**	**1.67**	**1.00**
Modification / Style	1.25	4.75	7.50	2.67	1.50
Modification / Move	3.50	1.25	2.75	3.33	3.50
Interview / Move	2.00	1.50	2.00	2.00	1.00
Interview / Grid	0.25	1.75	0.25	0.33	1.00
Interview / Color	1.25	1.25	2.00	1.00	2.00
Remaining Codes	74.00	112.50	93.25	102.67	82.50
Average number of appl.	151.25	175.75	200.00	176.00	158.50

after discussing them or just after thinking about them. The corresponding code "Sketch" was applied in 16 out of 34 videos. In total this code was applied 30 times so some participants even created multiple sketches for the same task. Furthermore, we observed that a lot of labelling is done during modeling (830 code applications). The amount of code applications for labelling is not very far from the amount of code applications for creating elements, i.e., shapes and lines (1081) compared to the other code applications. This seems to be a difference between modeling and drawing / illustrating where usually the amount of text is way less than the amount of shapes.

C. The Insights

The goal of this study was to identify interaction patterns while modeling on a whiteboard. Overall it is interesting that the amount of time that is coded (4 hours and 34 minutes) in the videos roughly half the time that is uncoded (7 hours and 52 minutes). That means that the participants spent half as much time actively interacting with the whiteboard as they spent not interacting with it. These insights are described below. Due to the limitations of a whiteboard, such as the lack of drag & drop or copy & paste, we would have expected participants to spend more time working on the whiteboard. From our study we have gained several insights.

1. Preferences for text behavior Preferences for text behavior were discussed in ten interviews but there is no clear preference among the participants whether text should defer to the size of the shape (i.e. the text wraps or its size shrinks to make it fit into the shape; coded as Interview / Text Defers) or text should dominate the size of the surrounding shape (i.e. when the text becomes too long / large the shape should be resized to make enough space for the text; coded as Interview / Text Dominates).

Discussion Of course the size of shapes or the size of font can be constrained by a concrete syntax but when this is not the case users of modeling tools should be able to set up their preference on this behavior.

As there does not seem to be a clear preference for either behavior, tools should enable their users to set up their preference for text behavior. There are tools like Microsoft PowerPoint that already have a setting for this preference but we do not know of modeling tools with such a setting or at least do not find it there.

2. Sketching In 16 out of the 34 videos we have observed participants drawing sketches (coded as Sketching). The sketches had different purposes. Sometimes they were used as means for communication and sometimes they were used to decide between ideas on how to approach the problem. Even in two interviews the participants said that if they had created a sketch beforehand they could have avoided some bad decisions.

Discussion From this observation we conclude that sketching seems to be a relevant aspect of modeling. There also is research dedicated to sketching [11] so sketching does also have direct relevance in research. As a consequence we think that tools should avoid being strictly correct by construction to allow for creating sketches. These sketches could either be free-hand drawings, images of sketches or even sketches using predefined shapes.

3. Consistency In 9 out of the 34 interviews the participants wished to have a tool that supports a consistent appearance of their model (coded as Interview / Consistency). In particular participants want to be able to define a shape and reuse the same shape all over their model. But there are different degrees of consistency wished by participants. There were participants that wanted all shapes to be exactly the same shape having the same size and style but there also were participants that just wanted to use the same shape with differing size or style. An example is one participant that wanted to draw a shape and copy & paste it to all the other positions.

Discussion In modeling tools this aspect is usually already enforced by the concrete syntax but in tools like Microsoft PowerPoint it is either possible to copy & paste elements or to define them in the so called slide master. A similar behavior is implemented in FlexiSketch [18]. In FlexiSketch users can create free-hand drawings and map them to elements of the meta-model. After creating such a mapping, the user can draw a similar shape and FlexiSketch suggests to transform it to the shape defined previously.

Obviously, there is no point for modeling tools to support defining own shapes as long as that is supported by the used concrete syntax. However, for example making the size of all shapes similar would be beneficial and it might also help for sketching.

4. Highlighting In 2 of the interviews there were participants that wanted to use colors to highlight specific parts of their model. The participants wanted to mark elements as being of higher relevance than others or as being different from the rest.

Discussion We can think of several different scenarios

Table V
WISHES OF PARTICIPANTS MENTIONED IN THE INTERVIEWS

Code	Description (The participants...)	Number of interviews with this wish
Interview / Move	wished they had a way to move around elements on their canvas	24
Interview / Color	wished they were not limited in the selection of colors they had	19
Interview / Size Ratios	wished they were assisted in having correct size ratios of lines	14
Interview / Grid or Ruler	wished they had a grid or ruler to arrange their elements to	12
Interview / Infinite Canvas	wished they had an infinite canvas	10
Interview / Consistency	wished they had a way to ensure consistency in their model	9
Interview / Resize	wished they had a way to resize elements	9
Interview / Multiple Diagrams	wished they could combine multiple diagrams on their canvas	8
Interview / Draft	said they would erase their model and redo it properly	7
Interview / Map	wished they had a map of the city	6
Interview / Bad Automation	wished tools avoided bad / unpredictable automation	5
Interview / Text Defers	wanted the text to defer to the size of the shape	5
Interview / Text Dominates	wanted the text to dominate the size of the shape	5
Interview / Auto Adaptions	wished there were automatic adaptions such as lines avoiding text	5
Interview / Dynamic	wished they had simulate their model	4
Interview / Copy & Paste	wished they had a way to copy & paste elements on their canvas	4
Interview / Zoom	wished they could zoom in / out on their canvas	3
Interview / Templates	wished they had templates for elements that they could reuse	3
Interview / Cut & Paste	wished they had a way to cut & paste elements on their canvas	3
Interview / Recolor	wished they had a way to recolor all elements of a selected type	2
Interview / Sketch	wished they had a way to pre-sketch their model before-hand	2
Interview / Import Data	wished they had a way to import data from a file	2
Interview / Bridges	wished they to have bridges where edges cross each other	2
Interview / Highlight	wished they had a way to highlight specific elements	2
Interview / Hyperlinks	wished they had a way to hide information behind clickable text	2
Interview / Layers	wished they had multiple layers to work with	1
Interview / Identify Errors	wished they had automated validation for errors	1
Interview / Oriented Lines	wished they could align their lines to the edges of the canvas	1
Interview / Dashed Lines	wished they had a tool to directly draw dashed lines	1
Interview / Undo	wished they had a way to undo performed actions	1
Total number of interviews		34

where highlighting specific elements can be useful. In one of these scenarios highlighting is used for indicating incomplete parts similar to using TODO notes in textual tools. In another scenario it is used for marking specific parts that should be looked at by someone else.

We conclude that tools should provide a way to highlight certain parts, as there are several use cases for this feature.

5. Bad Automation In 5 out of the 34 interviews there were participants that complained about tools having unpredictable bad automation. For example one participant said that unlike from what he has seen in some tools, a tool should never change font size automatically. Another participant said that automatic layouting can be great but that it should still provide full control over the layout. It seems like participants generally prefer defensive automation, i.e., only automatically adjust where it is certain that the adjustment is what the user wants, over aggressive automation, i.e., always try to adjust everything to please the user.

Discussion We conclude that tools should try to be less aggressive on automation and accept that they cannot predict all adjustments a user wants.

6. Education Fields Comparing the results of the different education fields (see Table IV), the participants of the group "Mathematics" rather used rectangles than circular shapes.

Discussion It is difficult to find an explanation for this but it might well be possible that having one more pair in Mathematics would alter this ratio. The students of computer science tended to write the label first and surround it with the shape afterwards. This could be explained by them being more experienced with modeling and trying to avoid the need to resize the shape afterwards (see the Labelling codes prefixed by Labelling / Inside).

7. Colors In 19 out of 34 interviews the participants complained about having too few different colors (coded as Interview / Color). The participants used colors to distinguish different semantics for similar looking shapes or to visually group labels with the same semantics.

Discussion From this result we conclude that tools should provide multiple colors. In our study the participants had four different colors and complained about having too few. That means a tool should ideally provide more than four different colors.

8. Undo In only 1 out of 34 interviews the participants wished they had undo functionality.

Discussion We are surprised about this number being so low. We would have expected it to go in line with the top codes. Maybe on a whiteboard people do not think in actions or transactions which could be undone and because of that do not think about undoing.

We are convinced that undoing is an essential and well

established feature so we do *not* conclude from this result that undo would be unnecessary.

V. Threats to validity

In this section we discuss how we tried to ensure validity and what threats to validity we are aware of. We consider the threats described by Runeson and Höst [19].

Construct Validity To ensure construct validity we had a pilot run. In this pilot run everything went as desired besides that the tasks were a bit too long. With the reduction of the tasks a logical mistake was introduced into Task 2. In this task the participants had the daily schedule of a canteen kitchen and in the description of a task there was a reference to a previous task with time different to the time specified for the previous task. This mistake is a threat to construct validity. One participant pointed this mistake out to us, a few other participants asked us if they understood it correctly but in the end every participant just ignored the inconsistent description and used the correct time that was specified for that task. So in the end this mistake only caused a bit of confusion which slowed down the modeling at this point but it did not affect decisions.

Internal Validity A threat to internal validity is that most of the students that fell into the group of Design are students of media informatics. The acquisition of participants was complicated because we always had to find two people with similar education that have to have time to spare in the same time slots. Internal validity is improved by asking the participants about their decisions in the semi-structured interviews. Data triangulation (comparing interview answers to the actual behavior while performing the tasks) was used to increase internal validity.

External Validity External validity of a qualitative study design like ours is typically low. However to improve external validity, we used two different tasks and tried to find participants from different education fields. One threat to external validity is that the group of Mathematics has one pair less than the other groups and the group "Other" is missing even two pairs. That makes it more difficult to generalize the findings for these two groups and also complicates the comparison of the groups. To allow participants with different education we had to choose examples that everyone is familiar with. This additionally threatens external validity as the examples do not match what is typically modeled in software or systems engineering. But there are similarities to UML diagrams. The first task is structurally similar to component diagrams, i.e., stations and paths taken by buses or trains. Additionally, some participants used something similar to hierarchical states in a state machine. They had a large element containing the stations within the city. This large element was part of the train system with the other cities being elements of regular size.

Reliability To ensure reliability of our study we wanted to provide all raw data but unfortunately we did not get permission to provide the recordings. So we can only provide the raw coding data [20]. If the study is to be conducted with other participants, the creative freedom in the interviews could be a threat to reliability. While there were answers given by almost every participant there also were answers only by single participants. It will be easy to find other participants giving the answers that were given frequently by participants but it will be difficult to find other participants that will give the same answers as the ones that were unique in our interviews. Another threat to reliability is that all material is in German so it requires good knowledge of the German language to translate the material to another language. To improve reliability, we first coded the same video independently. While discussing the results we developed a common coding system. With this common coding system we coded another video independently to see how well we agree and how well the coding system works. After this run we had quite similar results. We discussed the differences we had and optimized the coding system a bit so we could finally code the rest of the videos independently.

VI. Conclusion and Future Work

This paper addressed an issue in the domain of software and systems engineering. There the usability of existing tools is often described as poor. On the basis of an empirical study we have gained eight insights into how modeling tools can be improved. As described above, these insights are: Preferences for text behavior, Sketching, Consistency, Highlighting, Bad Automation, Education Fields, Colors and Undo. This paper described the conducted study, what insights have been gained and how they have been gained. The gathered data can be further investigated, for example taking the order of interactions into account. Furthermore, the insights should be implemented in a tool and evaluated in another study to see how these insights can help to improve the usability of modeling tools. In that study it would be interesting to investigate how much the new features are used if they are used at all, how the improvements actually improve usability either in terms of efficiency by timing the operations or in terms of satisfaction by asking the users how it felt using the improved tool. For example, it would be of interest if people actually used a feature to draw sketches within their model. Or if they preferred to draw a sketch on paper and import a photo of it into their model. Additionally, it would be interesting to see if people want to keep the sketch embedded into their model for example for documentation or archiving purposes.

After the study each participant told us it was fun participating in this study. This is in contrast to what the participants told us after a study on evaluating the usability of Yakindu[1]. There the participants said they were frustrated afterwards. Unfortunately, we do not have data for this

[1] https://www.itemis.com/en/yakindu/state-machine/

because in both cases it was not recorded. Based on this informal feedback, we plan to run a replication of the study with colleagues at other universities where we additionally gather data on user experience and particularly on frustration and the reasons for it. Specifically, we are interested whether this is just a spurious correlation or that using a whiteboard compared to a modeling tool is a causative factor.

REFERENCES

[1] OMG, *OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1*, Object Management Group Std., Rev. 2.4.1, August 2011. [Online]. Available: http://www.omg.org/spec/UML/2.4.1

[2] ——, *OMG Systems Modeling Language (OMG SysML), Version 1.3*, Object Management Group Std., 2012. [Online]. Available: http://www.omg.org/spec/SysML/1.3/

[3] A. Bobkowska and K. Reszke, "Usability of UML Modeling Tools," in *Proceedings of the 2005 Conference on Software Engineering: Evolution and Emerging Technologies*. Amsterdam, The Netherlands: IOS Press, 2005, pp. 75–86.

[4] N. Condori-Fernández, J. I. Panach, A. I. Baars, T. Vos, and Ó. Pastor, "An empirical approach for evaluating the usability of model-driven tools," *Science of Computer Programming*, vol. 78, no. 11, pp. 2245–2258, 2013.

[5] G. Liebel, N. Marko, M. Tichy, A. Leitner, and J. Hansson, "Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice," *Software and System Modeling*, vol. 17, no. 1, pp. 91–113, 2018. [Online]. Available: https://doi.org/10.1007/s10270-016-0523-3

[6] H. Störrle, "How are conceptual models used in industrial software development?: A descriptive survey," in *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, EASE 2017, Karlskrona, Sweden, June 15-16, 2017*, E. Mendes, S. Counsell, and K. Petersen, Eds. ACM, 2017, pp. 160–169. [Online]. Available: https://doi.org/10.1145/3084226.3084256

[7] S. Branham, G. Golovchinsky, S. Carter, and J. T. Biehl, "Let's Go from the Whiteboard: Supporting Transitions in Work Through Whiteboard Capture and Reuse," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. ACM, 2010, pp. 75–84. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753338

[8] M. Cherubini, G. Venolia, R. DeLine, and A. J. Ko, "Let's Go to the Whiteboard: How and Why Software Developers Use Drawings," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 557–566.

[9] C. H. Damm, K. M. Hansen, and M. Thomsen, "Tool Support for Cooperative Object-oriented Design: Gesture Based Modelling on an Electronic Whiteboard," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '00. New York, NY, USA: ACM, 2000, pp. 518–525.

[10] J. Rooksby and N. Ikeya, "Collaboration in Formative Design: Working Together at a Whiteboard," *IEEE Software*, vol. 29, no. 1, pp. 56–60, Jan 2012.

[11] N. Mangano, T. D. LaToza, M. Petre, and A. van der Hoek, "Supporting Informal Design with Interactive Whiteboards," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 331–340.

[12] N. Mangano, T. D. LaToza, M. Petre, and A. van der Hoek, "How Software Designers Interact with Sketches at the Whiteboard," *IEEE Transactions on Software Engineering*, vol. 41, no. 2, pp. 135–156, Feb 2015.

[13] T. R. G. Green and M. Petre, "Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework," *J. Vis. Lang. Comput.*, vol. 7, pp. 131–174, 1996.

[14] M. Petre, *Reflections on Representations: Cognitive Dimensions Analysis of Whiteboard Design Notations*, ser. Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series, M. Petre and A. Van Der Hoek, Eds. CRC Press, 2013.

[15] E. D. Mynatt, "The Writing on the Wall," in *Human-Computer Interaction INTERACT '99: IFIP TC13 International Conference on Human-Computer Interaction, Edinburgh, UK, 30th August-3rd September 1999*, M. A. Sasse and C. W. Johnson, Eds. IOS Press, 1999, pp. 196–204.

[16] Qi Chen, J. Grundy, and J. Hosking, "An e-whiteboard application to support early design-stage sketching of UML diagrams," in *IEEE Symposium on Human Centric Computing Languages and Environments, 2003. Proceedings. 2003*, Oct 2003, pp. 219–226.

[17] K. Karrer, C. Glaser, C. Clemens, and C. Bruder, "Technikaffinität erfassen–der Fragebogen TA-EG," in *Technikaffinität Erfassen–der Fragebogen TA-EG*, ser. Der Mensch im Mittelpunkt technischer Systeme, 2009, no. 8, pp. 196–201.

[18] D. Wüest, N. Seyff, and M. Glinz, "Flexisketch: A lightweight sketching and metamodeling approach for end-users," *Softw. Syst. Model.*, vol. 18, no. 2, pp. 1513–1541, Apr. 2019. [Online]. Available: https://doi.org/10.1007/s10270-017-0623-8

[19] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, p. 131, Dec 2008. [Online]. Available: https://doi.org/10.1007/s10664-008-9102-8

[20] M. Stegmaier, "Raw coding data for "insights for improving diagram editing gained from an empirical study"," Jul 2019. [Online]. Available: https://www.doi.org/10.5281/zenodo.3270564