Year: 2015

Sketching and Notation Creation with FlexiSketch Team: Evaluating a New Means for Collaborative Requirements Elicitation

Wüest, Dustin ; Seyff, Norbert ; Glinz, Martin

Abstract: Whiteboards and paper allow for any kind of notations and are easy to use. Requirements engineers love to use them in creative requirements elicitation and design sessions. However, the resulting diagram sketches cannot be interpreted by software modeling tools. We have developed FlexiSketch as an alternative to whiteboards in previous work. It is a mobile tool for model-based sketching of free-form diagrams that allows the definition and re-use of diagramming notations on the fly. The latest version of the tool, called FlexiSketch Team, supports collaboration with multiple tablets and an electronic whiteboard, such that several users can work simultaneously on the same model sketch. In this paper we present an exploratory study about how novice and experienced engineers sketch and define ad-hoc notations collaboratively in early requirements elicitation sessions when supported by our tool. Results show that participants incrementally build notations by defining language constructs the first time they use them. Participants considered the option to re-use defined constructs to be a big motivational factor for providing type definitions. They found our approach useful for longer sketching sessions and situations where sketches are re-used later on.

DOI: https://doi.org/10.1109/RE.2015.7320421

Originally published at:
Wüest, Dustin; Seyff, Norbert; Glinz, Martin (2015). Sketching and Notation Creation with FlexiSketch Team: Evaluating a New Means for Collaborative Requirements Elicitation. In: 23rd IEEE International Requirements Engineering Conference (RE'15), Ottawa, Canada, 24 August 2015 - 28 August 2015. IEEE, 186-195.
DOI: https://doi.org/10.1109/RE.2015.7320421

Sketching and Notation Creation with FlexiSketch Team: Evaluating a New Means for Collaborative Requirements Elicitation

Dustin Wüest
Department of Informatics
University of Zurich
Switzerland
wueest@ifi.uzh.ch

Norbert Seyff
University of Applied Sciences and Arts
Northwestern Switzerland, and University of Zurich
Switzerland
norbert.seyff@fhnw.ch

Martin Glinz
Department of Informatics
University of Zurich
Switzerland
glinz@ifi.uzh.ch

***Abstract*—Whiteboards and paper allow for any kind of notations and are easy to use. Requirements engineers love to use them in creative requirements elicitation and design sessions. However, the resulting diagram sketches cannot be interpreted by software modeling tools. We have developed FLEXISKETCH as an alternative to whiteboards in previous work. It is a mobile tool for model-based sketching of free-form diagrams that allows the definition and re-use of diagramming notations on the fly. The latest version of the tool, called FLEXISKETCH TEAM, supports collaboration with multiple tablets and an electronic whiteboard, such that several users can work simultaneously on the same model sketch. In this paper we present an exploratory study about how novice and experienced engineers sketch and define ad-hoc notations collaboratively in early requirements elicitation sessions when supported by our tool. Results show that participants incrementally build notations by defining language constructs the first time they use them. Participants considered the option to re-use defined constructs to be a big motivational factor for providing type definitions. They found our approach useful for longer sketching sessions and situations where sketches are re-used later on.**

***Index Terms*—Requirements engineering, collaboration, tool, sketching, ad-hoc modeling, notation definition, meetings,**

I. Introduction

Collaboration in Requirements Engineering (RE) often includes the usage of diagrammatic sketches to record and convey relevant information. Whiteboards and flip-charts are common tools used for brainstorming sessions to support requirements elicitation, design and idea generation [1], [2]. Creating notations ad-hoc during such a session allows to describe ideas at various levels of detail, and often leads to simple, but also ambiguous sketches [3]. While engineers can choose notations that can be understood by all participating stakeholders, these notations typically deviate from standards such as UML [2], [4]. Therefore, the created sketches might be hard to understand for stakeholders who did not participate in the session and do not know the context and intentions behind the sketches [5]. These stakeholders have to assume meanings for symbols which might lead to wrong interpretations. Even for meeting participants themselves it can be challenging to correctly interpret sketches a few weeks later [1], [3]. To re-use sketches during the RE process, engineers either take photographs and include them as non-editable files in other documents, or they manually build formal models from scratch based on the sketches, which can be a time-intensive task [1].

In previous work, we presented our first FLEXISKETCH prototype, a tablet-based tool for free-form sketching and the creation of node-and-edge diagrams [6], [7]. In contrast to traditional sketching tools, it is possible to specify the sketched constructs by, e.g., assigning types to them. Based on this information the tool infers a simple metamodel while the user is sketching. Being able to freely sketch models and at the same time creating a custom modeling notation on the fly allows to export and re-use the sketched models and metamodels in other modeling and metamodeling tools.

We have now extended our tool solution to support collaborative sketching and notation creation. FLEXISKETCH TEAM features synchronous, co-located, and multi-display collaboration. The technical details are described in [8]. Our approach allows multiple users to edit the same sketch concurrently using their own tablets. Users can also collaboratively define a modeling notation on the fly and re-use this notation in later RE sessions. They are free to choose notations that are comprehensible by all involved stakeholders, and can define a simple metamodel for it. To our knowledge, our approach is unique in the sense that it supports ad-hoc metamodeling in a collaborative sketching environment. This opens up interesting research opportunities regarding collaborative metamodeling.

The contribution of this paper is our conceptual solution for FLEXISKETCH TEAM, followed by an explorative study about how teams collaboratively define notations on the fly when supported by our prototype. The study includes two qualitative experiments: a laboratory experiment with students, and simulated workshop meetings with practitioners. In contrast to studies about collaborative sketching (e.g., [2], [9], [10]), we wanted to investigate when and how teams define their notations while sketching, if the resulting notations are consistent, and whether all team members participate actively in defining the notations.

978-1-4673-6904-6/15/$31.00

RE 2015, Ottawa, ON, Canada
Research Paper

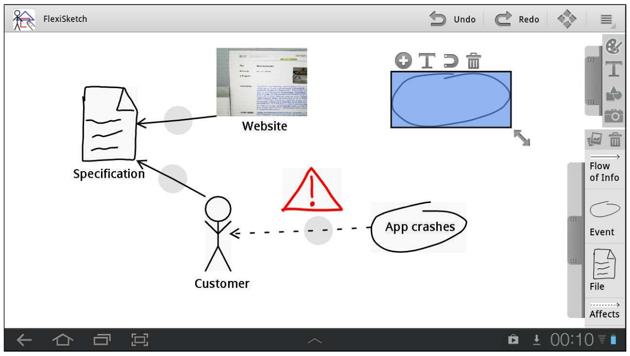

Fig. 1. Screenshot of FlexiSketch showing the UI and a model sketch.

II. FlexiSketch

The single-user version of FlexiSketch is an Android tool for model-based sketching (see Figure 1) [6]. The main idea of our tool is that users can freely and seamlessly interleave sketching and metamodeling tasks (i.e., defining a syntax for sketched symbols and links). Users can sketch freely as well as draw models with a defined notation (that users may create themselves), including any combination of the two options.

Strokes from the user are converted into a distinct symbol when the user lifts the finger for a specified amount of time. If a user draws a stroke from one symbol to another, the stroke gets converted into a link between the symbols. Existing images can be imported into the sketch and behave like symbols. Each element on the screen can be selected and moved around, and a context menu provides additional editing features (e.g., resizing, adding text, deleting).

Users can also assign (arbitrary) types to elements, and thereby define the vocabulary of a modeling language. Our tool manages multiple type libraries. A type library contains a list of all user-defined types from the current sketch, together with their visual representations. Type libraries can be stored and loaded independently from sketches, and can be changed at any time. All types of a type library are shown at the right edge of the screen, from where users can re-use elements via drag&drop. Alternatively, a sketch recognition algorithm detects drawn symbols that resemble defined types. The tool infers cardinality rules for links and automatically builds a metamodel according to the sketch and type library definitions. More details about the metamodeling features and a step-wise formalization of model sketches are provided in [7].

III. FlexiSketch Team

In this section we describe our novel tool version that supports collaborative work[1]. Our envisaged usage scenario consists of brainstorming and design sessions in RE where the participating requirements engineers and other stakeholders collaborate in creating ideas, eliciting requirements, designing solutions, and negotiating viewpoints. We focus on co-located settings where communication between participants, apart from sketching, happens via natural language and gestures. As our tool allows for arbitrary node-and-edge diagrams, it is not only suited for RE, but also for a broader software engineering context. However, we focus on RE sessions as described above, because we believe that these are the sessions where informal diagram sketches are most frequent [1], [3]: early in the software process, and when outside stakeholders (e.g., customers not knowing UML) are present.

A. Design Considerations

Analyzing our first tool version [6] and related work, we identified five key design issues (D) for the collaborative tool. These considerations also reflect selected design guidelines reported in research about computer supported collaborative work [10], [11], [12], [13].

D1. *To foster active participation, all meeting participants should be able to concurrently sketch on the drawing canvas and define notations.* This allows to work in parallel and save time in writing down information [10]. Further, participants should be able to choose where they stand or sit, while still having direct physical access to the workspace [13].

D2. The tool needs to *prevent conflicting inputs of participants*. It especially has to make sure that users can not concurrently change the defined notation in contradictory ways. This is closely related to coordinating the actions of participants as mentioned by Gutwin and Greenberg [12].

D3. The tool should *provide both shared and private views*. As it is harder to keep a shared focus when parallel work is supported [10], a shared view can mitigate this problem and helps participants to be aware of each other's activities [11]. In addition, participants should have private views that they can manipulate. If these views are extended to private workspaces, users can take notes that are not shared with the group [13].

D4. *Results of a design session should be provided immediately to all participants*. Everyone should be able to leave the meeting in possession of the diagram sketches and the defined notations. Meetings exist "as part of a larger context of overarching activities" [13], and therefore some of the created information is likely to be re-used.

D5. The tool should *increase the awareness of each other's actions* by enabling participants to monitor each other [12]. Without this support, it might be hard to tell who is doing what; especially when a user manipulates a diagram element via its context menu, and the resulting effect is perceived by others only when the manipulation already finished [11].

Due to time constraints, addressing issues D1 to D4 had to be given priority for our study about collaborative notation definition. A locking mechanism (see Sect. III-B) that tackles D2 also partially addresses D5. This solution was sufficient for conducting our study. However, we will implement further awareness features in the future.

B. Technical Solution

Our novel tool version addresses design issue D1 by providing a multi-screen setup where all workshop participants have tablets and concurrent editing access to a synchronized canvas (see Figure 2). Participants connect their tablets to a server via an ad-hoc wifi network. The server is a computer running

[1] A demo video is available at http://youtu.be/0kHjNfHLViM

Fig. 2. Meeting participants collaboratively create and discuss a model sketch using multiple tablets and a synchronized drawing canvas.

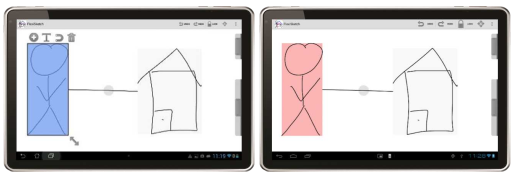

Fig. 3. Left tablet: a symbol is selected and appears in blue. Right tablet: the symbol is locked and appears in red.

FLEXISKETCH DESKTOP, a desktop version of our tool. This is a standalone version that is compatible with electronic whiteboards. Therefore, users can choose to work with tablets, on an e-whiteboard, or both. Alternatively, the server can be connected to a normal projector and shows an overview of the sketch canvas and a list of all defined elements. If not used actively, the desktop version automatically zooms its view to always show the whole sketch, while each participant scrolls and zooms his/her own view on the tablet (D3).

As soon as strokes from a user get converted into a distinct element, or a user performed a manipulation on an existing element, this element gets synchronized across the tablets and the server (D1). Similarly, any changes to the type library are synchronized immediately. This also means that participants will leave the room with the meeting results on their personal tablets – without additional effort (D4).

With FLEXISKETCH TEAM, multiple users can sketch simultaneously within the same canvas region, and can define types of different elements concurrently. If the same type gets assigned to two different elements, the tool generates only one type entry, but stores both elements as alternative representations for that type. A non-optimistic locking mechanism [14] prevents inconsistent states of individual elements by prohibiting the concurrent editing of the same element: the context menu of an element is accessible by only one user at a time (D2). Otherwise a user could, e.g., delete an element while another user is in the middle of adding text or assigning a type to it. The server locks an element when a user selects it. On all other tablets, this element is then shown with a red background and does not react to user inputs (Figure 3). The user can de-select the element by tapping on any other part of the sketch canvas, whereupon the server unlocks the element. Using visual cues to show locked elements also provides some user awareness in the sense that users can see what model parts are currently edited by others. This is a first step towards D5.

A share function makes it possible to push the current sketch and notation from one tablet to the other tablets and the server. Therefore, meeting participants can, e.g., i) prepare different ideas before the actual meeting and then share and discuss them in the meeting, ii) disconnect their tablets during a meeting to have a private workspace, and then re-connect to share their work and ideas (D3).

IV. STUDY GOAL AND METHOD

The goal of our explorative study was to investigate how requirements engineers (both novices and experienced practitioners) collaborate in a workshop setting when supported with FLEXISKETCH TEAM. While the focus is on the collaborative definition of notations, analyzing the sketching behavior as well provides the necessary context. We refined our goal in three research questions:

Q1: How do collaborators sketch when they are provided with a collaboration tool that supports simultaneous sketching on multiple screens?

Q2: How do collaborators define and agree on a common modeling language and notation when they sketch?

Q3: What are the benefits and limitations of our tool-supported collaboration approach as perceived by the collaborators?

We conducted a laboratory experiment with graduate students (i.e., novices), followed by an observational study with practitioners in a simulated workshop setting. We included students for two reasons: (i) we wanted to test our approach with both novice modelers and experts, in order to assess how novices cope with our approach, and to identify potential differences to experts. (ii) The student experiment also served as a test whether our tool prototype was good enough for showing it to industrial practitioners.

For the experiments, every participant received an Android tablet with the tool installed. The tablets had capacitive screens with sizes ranging from 9.4 to 10.1 inches. Participants could choose to use their fingers or a stylus. While we were not able to provide identical tablets for all participants, we believe that this reflects a real-world scenario where engineers bring their different, personal tablets to a meeting. We decided not to use an e-whiteboard for the study, as this allowed us to be more flexible where to perform our study (travel farther) and extend the amount of potential study participants. Instead, we used the desktop version of our tool to provide a shared view displaying the overview. The study was conducted in German. Quotations presented in this paper were translated to English.

A. Laboratory Experiment

The experiment was incorporated into an advanced requirements engineering course in a Swiss university, but we made clear that the students' performance in the experiment does not influence their grades. Eight graduate students in computer science were visiting the course, some of them having several years of industrial experience. One student already knew an old version of our tool because he had participated in an early usability study. The course size allowed us to form three groups. Group G1 consisted of students S1 and S2, groups G2 and G3 had students S3-S5 and S6-S8 respectively. We found this to be a realistic group size for the kind of ad-hoc meetings that we want to support with our tool. The students already knew each other from solving group homework.

Our tool was introduced via a short training session before the experiment. We explained the main features of the tool and gave the students about five minutes to try out our tool in single-user mode. For the actual experiment, each group sat around a table. A computer, running the desktop version of our tool, was placed at each table and displayed the overview.

We gave the students two tasks and instructed them to solve these collaboratively within the groups, but we did not say how. The first task was to draw a use case (UC) diagram[2] for a web platform where students can share all kinds of documents. The second task was to create a user interface mockup (GUI) for the use case "sign up on the online portal". Both tasks were given in written form (in natural language) and included a prompt to be creative and depict as many ideas as possible, as well as to assign types to all elements on the sketch canvas. During the experiment, there was a supervisor assigned to each group who did not become active unless there was a technical problem. The students had ten minutes for each task, and the time was controlled by the supervisor.

At the end, each group had a discussion about the experiment for five to ten minutes, moderated by the supervisor. In addition, students were asked to fill out an online survey after the session[3]. Seven students filled out the survey.

The experiment data we collected and analyzed includes video recordings of each group, FLEXISKETCH log files listing user actions with timestamps, and participants' feedback from the discussion and survey.

B. Simulated Workshops

We organized three simulated workshop sessions with three requirements engineers per workshop. Again, we deemed this to be a realistic group size for ad-hoc meetings, and we wanted to keep a consistent group size over both experiments. Group G4 (practitioners P1-P3) consisted of practitioners from different companies in Switzerland, who are friends from their time at the university. The members of group G5 (practitioners P4-P6) work together in a university setting in Austria, but regularly deal with real-world problems from industrial partners. Practitioners P7-P9 from group G6 work together within a company in Austria. P4 and P8 are one hierarchy level above their co-workers. Other than this, we did not identify any power relationships. All practitioners except P2 did not know our tool before the workshop.

We introduced participants to the single-user version of our tool in a short training session (five to ten minutes) at the beginning. Then we asked the participants to think about a current RE related task or problem from their organization. This was subsequently used as collaborative ideation and modeling task within the simulated workshop.

We then introduced the participants to the collaboration features. A projector was used to display the overview. We did not introduce the concept of a workshop moderator, because we wanted to see how participants organize themselves using our tool. We let the participants choose the seating themselves, in order not to influence their collaboration behavior.

The FLEXISKETCH meeting sessions, which were video recorded, were limited to 20 minutes. There was no interaction between the experiment supervisor and the practitioners during the sessions unless technical problems occurred. Semi-structured interviews concluded the sessions. The interviews also included the questions that we used in the student discussions and survey.

V. ANALYSIS

One of the authors analyzed each video in two iterations. During the first iteration, he coded the editing behavior of each participant with a binary function (*1* if participant is currently touching the tablet, else *0*). Smoothing was applied by mapping the data to a function of discrete, two-seconds time steps in order to leave out fine-scale structures while keeping the important behavioral patterns. In the second iteration, the author coded the conversation between the participants. Firstly, he created the coding scheme and conversation categories according to his experience from the first iteration, and then analyzed two videos (from student groups G2 and G3) to fine tune the scheme and categories. The outcomes were discussed with the other authors and research colleagues. After finalizing the coding scheme, the author processed the rest of the videos. For each participant and utterance, he coded whether the participant was speaking about the modeling language (*semantics*), the modeling task (*modeling*), tool-related subjects such as usability and specific features (*tool*), or topics unrelated to the task and tool (*other*). The semantics category includes utterances about the notation (e.g., "*What does this element mean?*", "*I'm going to draw and define an actor symbol*"). Modeling utterances are related to the domain model (e.g., "*We have a further actor, professor, who can also upload documents*"). Examples for tool utterances are "*Can symbols be rotated?*" and "*You need to hold down the finger to drag and drop*". Finally, an example for an unrelated utterance is "*You have nice drawing skills*". After the categories were created, we looked more closely at the semantics category to find out how participants communicated their type definitions (e.g., do they talk to their team members before or after creating a type? Do they discuss or just notify each other?).

Due to a software bug, we could not obtain complete logging data from all tablets. But where available, the tool logs were used for triangulating the video data. The data from the discussions, survey, and interviews was analyzed to gather data for Q3. Statements from discussions were grouped to find interesting patterns and recurring statements. We also looked for correlations between survey answers, discussion statements, and participants' behavior during the experiment.

[2]We predefined the problem and diagram types for the students in order to create a shared work context for each group, and because they were novice modelers – we did not want to risk overwhelming them with the creation of new modeling languages in the first-time evaluation of FLEXISKETCH TEAM.

[3]https://files.ifi.uzh.ch/rerg/flexisketch/StudentHandouts.pdf

VI. Results

A. Sketching and Collaboration Behavior

R1.1: Phases of simultaneous sketching happened in all groups. Figures 4 and 5 show when participants were talking and/or editing. All six groups revealed a working style where they had phases of silent, simultaneous editing and phases of discussions with and without editing. In the practitioner groups, all three group members were simultaneously editing during 13.5% (G4), 10.1% (G5), and 8.2% (G6) of the total session time. These values were higher for students with 51.5% (G1, the group of two), 20.1% (G2), and 23.3% (G3). This difference between practitioner and student groups correlates with the different amount of communication (see R1.2). Practitioner P4 (in G5) did not draw much, instead she helped by asking many explorative questions about the project and possible language constructs, e.g., "*Should we have different feature types or just one type called feature?*". We identified student S3 as a leader in G2. He talked the most and came up with many modeling ideas, while the other members concentrated more on actual sketching activities. No clear leader emerged in the other groups.

R1.2: Practitioners communicated more than students. Practitioners were talking during 351 (G4), 415 (G5), and 314 (G6) of the discrete two-seconds time steps, which results in a mean of 12 talking minutes per group, while students talked during 203 (G1), 234 (G2), and 253 (G3) time steps, resulting in a mean of 7.7 talking minutes. Practitioners from all groups stated in the interview that there were no communication issues while working with our tool, and no group members disagreed. In contrast, students from G1 and G3 stated that their attention was drawn to the interaction with the tool. They believed that this reduced the amount of discussions they had, e.g., S2 said: "*Especially at the beginning we did not talk, each of us was concentrating on his own tablet*", and S3: "*Each of us drew something. We only discussed after noticing that two of us had sketched the same thing and we needed to agree about what to keep and what to delete*".

It rarely happened that a practitioner drew something without notifying the others about it. One exception happened in G4: At the beginning of the session, practitioners discussed every step before sketching something (e.g., P1: "*I'm going to draw a system boundary, okay?*"). Towards the end, communication regarding planned actions started to decrease: they were simultaneously sketching three different types of diagrams next to each other. They ensured consistency between diagrams by discussing key elements which were important for all diagrams, such as specific stakeholders and use cases.

No student group started with a brainstorming or extended discussion. Instead, communication happened rather "incremental": multiple times during the session, they quickly mentioned ideas about what they could draw next and who will draw what parts, and then continued to draw silently.

R1.3: Participants tried to fit the whole sketch on their tablet screen. The diagram created by the practitioners in G6 fits on a tablet screen, while the diagrams from G4 and G5

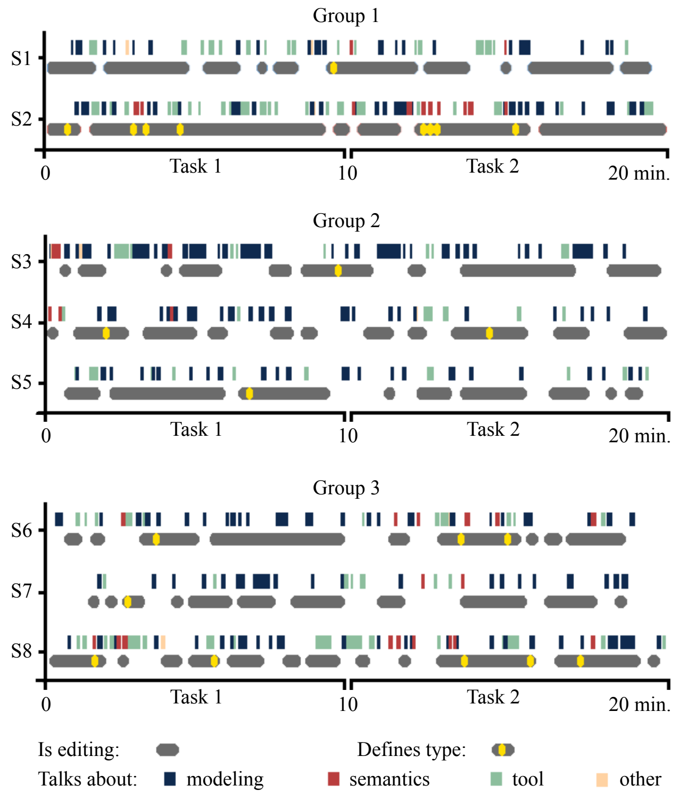

Fig. 4. Phases of editing and discussions in student groups.

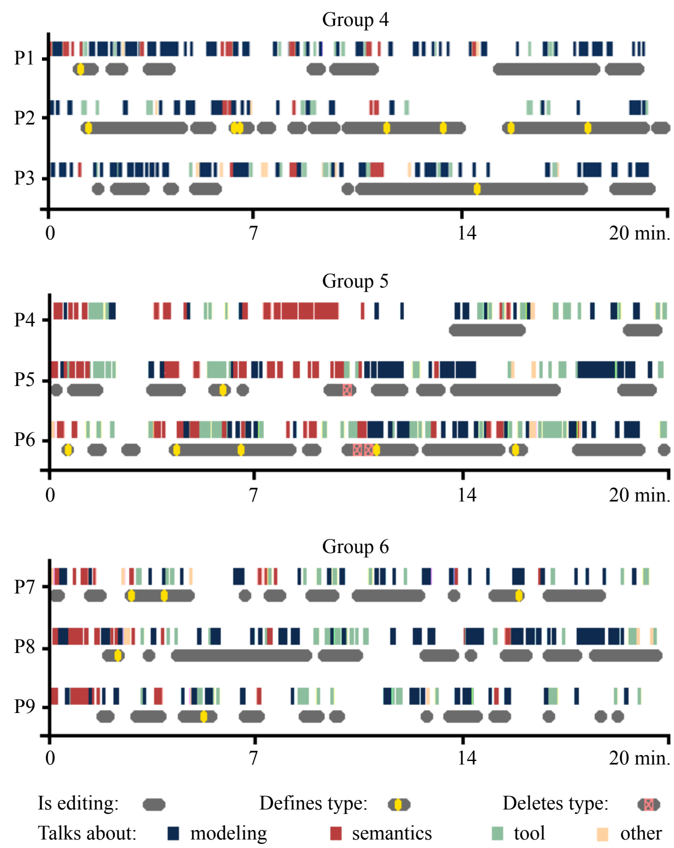

Fig. 5. Phases of editing and discussions in practitioner groups.

TABLE I
THE AMOUNT OF SYMBOLS, LINKS, AND DEFINED TYPES CONTAINED IN EACH DIAGRAM FROM STUDENTS (LEFT) AND PRACTITIONERS (RIGHT).

		#symbols	#links	#types
G1	UC	12	7	5
	GUI	13	0	4
G2	UC	10	6	3
	GUI	3	0	1
G3	UC	7	5	4
	GUI	8	0	5

	#symbols	#links	#types
G4	20	16	9
G5	18	15	3
G6	9	3	5

clearly extended that size. In contrast, all student groups made their diagrams fit on a single tablet screen (in a way such that no scrolling or zooming of the canvas was needed). S5 from G2: "*We wanted to make sure that we always see the changes made by each other, and that no change happens outside of a tablet's current view*".

The usage of the big screen with the overview varied significantly between groups: G4 and G6 barely looked at it. P1: "*We used it once or twice*". In contrast, interview feedback and the video from G5 revealed that they used to look at the big screen when discussing the design and further steps. Similarly, five students stated in the survey that the big screen with the overview was useful (see Table II).

R1.4: Participants peeked onto each other's tablet. All student group members were sitting close together, and all students took a look at others' tablets from time to time. Also, practitioners P1 and P2 in G4 and all three practitioners in G6 used to peek onto each other's tablet. P1: "*It helps to coordinate, to see what the other person is doing*".

B. Collaborative Notation Definition

R2.1: Notations were defined by multiple participants. The student groups defined a total of 9 (G1), 4 (G2), and 9 (G3) types, the practitioner groups defined a total of 9 (G4), 3 (G5), and 5 (G6) types. In all groups, type definitions were created by more than one participant (indicated by yellow dots in the sketching bars of Figures 4 and 5), with P4 being the only person who did not define any type.

For the student groups G1-G3 and practitioner group G4, the video analysis revealed that there were no discussions about the graphical representations of types, with one exception in G3 where S6 stated that he was about to declare a drawn symbol as *Use Case*. S8 intervened by asking him whether they should use a nice geometrical shape instead of the hand-drawn one, and S6 agreed.

In contrast, practitioner groups G5 and G6 briefly discussed in advance how the individual symbols representing the concepts should look like (see R2.4 for discussion details).

R2.2: Notations were defined incrementally during the whole sessions. All groups defined types whenever they introduced new elements in the diagram. Practitioner groups revealed a pattern where they discussed many semantics concerns in the early phase of the modeling task (especially G5 and G6, see Figure 5), followed by incremental discussions and ad-hoc notation definitions during the whole task.

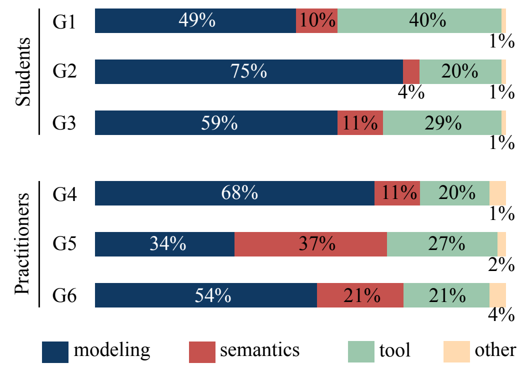

Fig. 6. Talk category distribution in student and practitioner groups.

R2.3: Participants based their notations on familiar concepts and symbols. Figure 8 shows defined types and extracts of the resulting diagrams from the practitioner groups[4].

All practitioner groups have chosen non-standard modeling notations loosely based on existing standards such as UML. Participants from G4 and G5 stated in the interview that they chose and agreed on concepts from languages that were familiar to everybody, and adapted them for use in their problem context. G4 used a notation which was very similar to UML, and most types were defined by P2 without discussions. The videos show that groups G5 and G6 started by discussing what types of diagrams they are going to draw, mentioning standard languages and important diagram elements. However, they then started to deviate from standards and introduced further concepts.

R2.4: Discussions about semantics depended on the chosen language constructs. In the practitioner groups, 11% to 37% of the total communication was devoted to semantics, depending on the group (Figure 6). G5 talked more about semantics than the concrete model, discussing a lot about how they can map their concerns to symbols. It was the only group that deleted some element types (Figure 5): in the middle of the modeling task, they discussed that three types have been defined at a too fine-grained level, and concluded to replace them by a more abstract type. In contrast, group G4 discussed not much about semantics. P1 said: "*Borrowing most of the elements from UML allowed us to get a shared understanding of the symbols' meaning with little effort*".

Figure 6 reveals that student groups talked little about the meaning of elements, i.e., the semantics. P2 mentioned: "*There was no need to discuss because we were all familiar with the use case diagram notation needed for the first task*". Figure 4 shows that G1 and G3 communicated more about semantics in task two (user interface), while G2 did not talk about semantics. Indeed, results show that G2 almost completely neglected type definitions for task two (Table I).

Figure 7 reveals how participants communicated their type definitions. In 40% of the cases (9 types), students talked about type assignments before they were actually doing them. In the other cases, they just informed each other by mentioning

[4]The full diagrams can be found in high resolution at https://files.ifi.uzh.ch/rerg/flexisketch/TeamResults.pdf

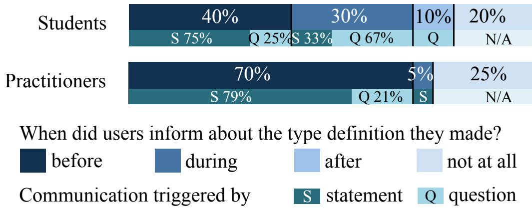

Fig. 7. When participants talked about new types – before, during, or after defining them.

the symbol type, either while inputting the types or only afterwards (this behavior was especially noticeable during the GUI diagram task, e.g., S6: "*Radio button*", S8: "*Text field*"). In these cases, 75% of the communication (7 types) was initiated by a team member asking another one what he/she is doing or what the definition means.

In contrast, practitioners discussed type definitions in advance in 70% of all cases (12 types). There were no questions related to a type definition after an element already got defined. In 20% of the cases, communication about a type definition was triggered in advance by a question (e.g., P8: "*Should we define a type named file?*"). The rest of the discussions were started by a statement.

Student groups did not discuss 20% of the type assignments at all. There was a similar high amount (25%) for practitioner groups. Especially G4, borrowing all element types from UML (except one type), defined four out of the nine types without any conversation.

R2.5: All groups created consistent notations. Table I shows the complexity of the resulting diagrams in terms of the number of defined types and elements drawn. The video analysis revealed that students from all groups always re-used existing types whenever possible. Therefore, all students within a group used the same notation: all resulting UC and GUI diagrams that we received from students showed a 1:1 correspondence between symbols and meanings. In practitioner group G4, two symbols with different types cannot be distinguished by the sketch recognizer (*'system'* and *'class'*). P2, the person who defined the latter, told us in the interview: "*I wanted to sketch a better class symbol, but refrained from it because the session was about to end*". Apart from this case, the resulting diagrams showed no inconsistencies in the notation (all symbols can be distinguished by form and/or color). There was no evidence from the video analysis and the log files that inconsistencies happened during the sessions.

Regarding completeness, the type definitions were complete apart from some exceptions: G6 did not define the *'trust boundary'* symbol, and student group G2 did not define GUI elements. No group except G5 defined links (see Figure 8).

C. Perceived Benefits and Limitations

R3.1: The drag&drop mechanism was frequently used. Table II shows that all students liked the drag&drop functionality for defined elements. S2 said: "*As soon as you start to make bigger sketches, dragging elements [from the type library] onto the canvas is faster than drawing them by hand each time*". The video analysis and the resulting diagrams confirm that all but G6 heavily re-used the defined types by using the type library's drag&drop mechanism. P8 from G6 stated: "*The possibility to re-use defined types is a big motivation for defining them*". During the experiment, P8 said: "*Can I rotate a symbol? ... No? ... In that case, I do not need to assign a type to this particular symbol*".

TABLE II
STUDENT ANSWERS REGARDING FLEXISKETCH FEATURES ON A LIKERT SCALE FROM "STRONGLY DISAGREE" TO "STRONGLY AGREE".

Activity / FLEXISKETCH feature	- -	-	o	+	++
Concurrent drawing was frequent	0	1	0	1	5
Modeling with the tool worked well	0	2	2	2	1
Many manipulation conflicts occurred	1	1	1	2	2
Lock mechanism was helpful	0	0	0	2	5
Lock is needed in same-place collab	0	1	2	3	1
Big screen was useful	0	1	1	1	4
I used drag&drop functionality	0	0	0	1	6
Drag&drop functionality was useful	0	0	0	0	7

R3.2: Defined types can serve as documentation. Practitioners from all three groups said they liked having the sketches immediately available in digital form after a FLEXISKETCH session. Furthermore, P5 and P9 stated that types assigned to symbols also serve as some kind of documentation and contribute towards the comprehensibility of a sketch. P5 said: "*Due to the type definitions, I think I will have less effort in understanding a sketch when I look at it again after several weeks or months*".

R3.3: Participants liked FLEXISKETCH. Table II shows student answers to selected questions from the online survey. All groups reported in the interview that they liked the ability to draw simultaneously. P7: "*The tool makes it easy for multiple persons to draw on a small region of the canvas*". Practitioner P8 added: "*If you are, for example, ten people and have three tablets, I think this would be enough. You can circulate the tablets, and the others [who currently don't have a tablet] can look at the overview on the big screen*". Student S6 stated: "*The tool allows to sketch multiple ideas at the same time. Afterwards the team members can discuss the different ideas*". Student S4 said that the multi-screen setting takes some time to get used to: "*It depends on the setting. If everyone is at the same place, I'd prefer a big screen that allows multiple persons to sketch physically next to each other. But in a distributed setting, FlexiSketch comes in handy*".

Four students reported that many manipulation conflicts occurred (i.e., multiple students wanted to manipulate the same element concurrently, which was prevented by the lock mechanism), while two students disagreed, and one was undecided. The video analysis confirms that concurrent manipulation did happen to different extents in the groups. Therefore, students stated in the survey that the lock mechanism is helpful. S5 added: "*Locked elements also provide visual clues about what the other group members are currently doing*".

Two students said in the discussion that they would have liked to have some kind of log, history, or color coding, in order to tell who has drawn what elements of the sketch. Two other students and two practitioners said that they would like

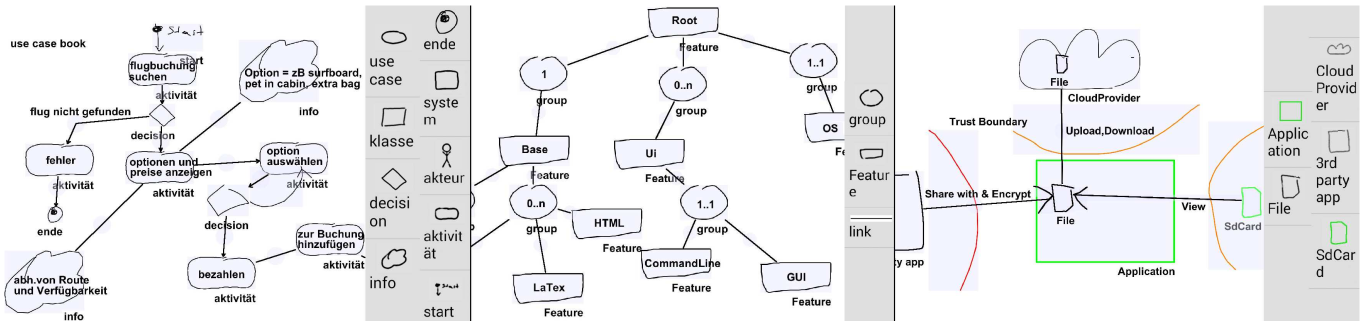

Fig. 8. Extracts from the results of practitioner groups (left: G4, center: G5, right: G6). The grey boxes show the defined elements.

TABLE III
A SUMMARY OF THE RESULTS, GROUPED BY RESEARCH QUESTION.

R1.1: Phases of simultaneous sketching happened in all groups
R1.2: Practitioners communicated more than students
R1.3: Participants tried to fit the whole sketch on a tablet screen
R1.4: Participants peeked onto each other's tablet
R2.1: Notations were defined by multiple participants
R2.2: Notations were defined incrementally during the sessions
R2.3: Participants agreed on familiar concepts and symbols
R2.4: Semantics discussions depend on chosen constructs
R2.5: All groups created consistent notations
R3.1: The drag&drop function was frequently used
R3.2: Defined types can serve as documentation
R3.3: Participants liked FlexiSketch
R3.4: Groups prefer FlexiSketch for large and re-usable sketches

to have an eraser function that allows them to erase only parts of symbols, as well as very small strokes that they made by mistake (currently, these strokes are hard to select and delete because of their small size).

R3.4: Groups prefer FlexiSketch for large and re-usable sketches. All groups said that they would prefer a classic whiteboard for coarse, short-lived and not too large sketches. With respect to size, the sketches created in the experiment were perceived to fall into this category. Only three students agreed that modeling with the tool worked well, while two were undecided and two were negative. They argued in the discussion that it is not worth dealing with some of the usability issues and having a less natural sketching feeling unless a sketch becomes bigger and exceeds the size from the experiment. Similarly, G4 and G5 stated that they would prefer our tool for larger sketches. P5: "*It will be easier to edit, store, and re-use them*". P1 said: "*FlexiSketch might unfold its advantages when officially introduced in a company and used for a prolonged time, over multiple workshops*".

VII. Discussion of Results and Design Implications

In this section, we discuss the results from the study (Table III provides a summary) and what they imply for the design of further collaborative sketching and notation definition tools.

Q1: How do collaborators sketch together? All participants took an active part in the sessions and used the possibility to sketch simultaneously (R1.1). Both students and practitioners took a look at others' tablets from time to time (R1.4), which can help in coordinating themselves (e.g., monitoring, assistance [12]). Loksa et al. [15] encountered the same phenomenon of "students peering onto the creator's tablet". R1.3 and R1.4 show that user awareness is important in a setting where multiple small screens can be used for input. P1: "*When sketching collaboratively with a tool, you can just start to draw. But here, when defining types, you need to be more careful and coordinate*". A separate, big overview screen can reduce the problem to a certain degree. However, our results suggest that it is preferable to show the overview on the same screen a user is working on. This leads to a tradeoff regarding screen space [11] and asks for new solutions regarding the small size of mobile devices. Mobility is an important advantage of our tool. But for non-mobile tools, a shared screen and view could lead to smoother collaboration.

Some students had problems to manage both the cognitive and the social space [16] at the same time (R1.2): they concentrated on the tool and did not communicate their actions well enough. This fits with a finding from Shih et al. [17] that users do not automatically "develop a sense of tolerance for lack of social awareness" in collocated sessions. However, studies suggest that it is possible to learn how to cope with a multi-space setting [15]. Indeed, we observed that practitioners did not have this problem and were able to coordinate their actions. This suggests that our tool needs additional awareness features to support less experienced users.

Q2: How do collaborators define and agree on a notation? Results R2.1 and R2.2 show that multiple participants in each team defined parts of the notation incrementally during the sketching task. All practitioner groups deliberately deviated from standard notations (R2.3). Dekel and Herbsleb found the same result [5]. Hence, discussions about semantics happened during the whole workshops (R2.4). While practitioners mostly communicated type definitions before they made them, students tended to perform the actions first, and talk about them afterwards. Compared to pure sketching environments, this collaboration style can lead to more confusion in our case because actions can also explicitly change the semantics. The student groups reported in the discussions that they noticed this and that they would probably focus more on their communication style if they receive a similar task in the future.

The drag&drop mechanism (that allows to re-use types) was heavily used and seems to have had a big effect on the notation definition behavior (R2.5, R3.1) and the consistency of diagrams. Firstly, it motivated participants to define symbol types right at the moment when they used them for the first time. Secondly, they re-used defined symbols whenever possible. In contrast to symbols, no group except G5 defined links. Possible reasons could be that link types cannot be dragged and dropped, and that FLEXISKETCH regards all links with the same appearance as being of the same (undefined) type, and therefore implicitly keeps a 1:1 mapping. Overall, our tool is an example of how a sketching tool can help to have consistent and unambiguous sketches at the end of a session if the users want this. A side-effect of the drag&drop functionality was that participants committed to notations early. Studies with physical media [5], [18] show that the meanings of symbols are re-discussed and changed during design sketching, which rarely happened in our case (a possible explanation could be that our experiment consisted of a single, and relatively short, session). Therefore, regarding creativity, it is an important design decision whether and in what form to include a type re-using mechanism in tools such as FLEXISKETCH. At least, users must perceive types to be easily changeable. A feature such as the typing mechanism can have both positive and negative effects: it can foster discussions about types and thus creativity, but it can also distract from the sketching task. S6 stated: "*The tool is great, but one also needs to think about a possible process. Maybe there could be a first meeting where participants only define the notation. And in the next meeting, participants can fully concentrate on the modeling task*".

Q3: How did collaborators perceive our approach? In general, our approach and tool features were very well perceived by participants (R3.1, R3.3), but they also mentioned minor usability issues and made clear that they would not use our tool in all situations (R3.3, R3.4). The practitioner groups reported to favor our tool for sketches that are, or will be, re-used (R3.4). Walny et al. [19] show that many sketches "undergo a variety of transitions" during software development. Sketches are re-used in different situations and contexts. Therefore, a flexible sketching tool should not just focus on supporting a single scenario (e.g., sketching in a workshop) or process step. Firstly, it should not impose a particular workflow on the user [18], and secondly, it should provide means for storing contextual information. Dekel and Herbsleb state that it is difficult to interpret artifacts without knowing the context in which they got created [5], which is especially true for (ambiguous) sketches. In that regard, practitioners stated that they also were motivated to provide type definitions because they can serve as means for documenting the sketched diagrams (R3.2). At the same time, R3.1 shows that some users only provide this kind of lightweight metamodel information if they get an immediate benefit out of it. Furthermore, tools should capture the history (i.e., traces) of who did what, and when. Teams do not want to have to write down this information manually [5].

VIII. THREATS TO VALIDITY

Conclusion validity. We conducted a qualitative study to get a first in-depth understanding how groups create ad-hoc notations. Quantitative studies are necessary in order to strengthen conclusion validity.

Internal validity. Participants were unfamiliar with the tool and its features for ad-hoc notation definition, which is a possible threat. To mitigate it, we gave an introduction to the tool. Yet, the desire of the participants to explore the new technology, as well as some minor usability issues, were potential distractions and could have influenced the collaboration task.

In a study like ours, participants might want to please the researchers by giving positive feedback. Therefore, we asked the students to fill out an online survey after the lecture, which allowed them to give feedback anonymously. The bias was mitigated for two practitioner groups by the fact that the second author was not involved in conducting the experiment, and it was only him who knew a contact person from the groups G5 and G6.

Construct validity. We asked students to create specific types of diagrams, which can influence the amount of discussions needed about semantics, as well as minimize usability issues since we already knew that these diagrams can be built with our tool. However, the lack of micro-coordination that was revealed in student groups does not depend on a particular modeling notation. Furthermore, it was not a potential threat in practitioner groups, because they tackled real-world problems and freely chose notations.

External Validity. The limited number of students and practitioners who were involved in our evaluation activities, as well as the limited geographical distribution (Switzerland and Austria) is a known threat (convenience sampling according to proximity). However, we involved both novice and expert modelers with different backgrounds and skills to strengthen external validity. During the 20-minute sessions, we identified collaboration patterns that confirm the usefulness of our FLEXISKETCH approach. The generalizability to longer sessions has yet to be verified.

IX. RELATED WORK

In requirements and software engineering, collaboration is often researched in the context of design [15], [20] and user interface creation [15], [21], [22]. Collaborative sketching is an important method to foster creativity and discuss design ideas [23], [3], [24]. To better understand the creative activities in software design, researchers studied how and why engineers use physical media (paper, whiteboards) specifically for software design, e.g., [1], [3], which motivated us to conduct research about more flexible modeling tools. Other researchers are more focused on understanding the behavior and low-level collaboration patterns of participants when working with physical media, e.g., [10], [12]. The findings resulted in design guidelines for software tools that support collaborative work [10], [11], [12], [13]. We connected the requirements for FLEXISKETCH with these guidelines to come up with a collaborative version of our tool. There are many software tools that support collaborative sketching and design work (e.g., Calico [2], The NiCE Discussion Room [13]). Settings with such tools can result in different collaboration behavior compared to physical media (e.g., because workspace awareness differs). Therefore, the influence of software tools on collaboration and sketching behavior has been studied in e.g., [2], [9], [13], [25].

While we also looked at collaborative sketching behavior when using FLEXISKETCH, the main focus of our study was to investigate how requirements engineers collaboratively define notations. Related work on this subject is still scarce. One reason is that, from a metamodeling perspective, it was long believed that metamodeling should only be done by metamodeling experts [26]. Indeed, it has been shown that end-user metamodeling is hard to achieve [27], [28]. In contrast, we concentrate on lightweight metamodeling (or "just enough metamodeling") for creating ad-hoc notations in an end-user friendly way (e.g., for requirements engineers and domain experts). This scenario leads to the question how teams decide and agree on notations. Dekel and Herbsleb [5] performed an observational study to find out what kind of notations are used in object-oriented design, and how they evolve during sessions. Ossher et al. [18] investigated notations used in software design sessions to conclude whether their flexible modeling approach can provide appropriate support. Both studies used physical media in the sessions. In contrast, our study investigates how non-expert metamodelers choose and define notations when using a flexible software tool.

Compared to other studies such as e.g., [17], we do not primarily focus on the quality of the results, but we are interested in evaluating the behavior of the participants in terms of micro-coordination [16] during notation creation.

X. Conclusions and Future Work

In this work we presented a qualitative study about how requirements engineers sketch and define ad-hoc notations collaboratively when supported by a flexible modeling tool. Our multi-screen, node-and-edge diagram sketching tool allows users to define custom notations on the fly by assigning types to elements. The qualitative study indicates that the tool fosters interleaving of sketching and type-defining activities, and motivates all group members to perform both activities. Users managed to define consistent notations for their sketches collaboratively and reached a common understanding of the respective notations.

Results such as R1.3 and R1.4 suggest that having additional awareness features in the tool (knowing what the other users are doing) would be beneficial. In our future work, we plan to improve FLEXISKETCH according to these results. We also plan to perform longitudinal evaluations in industrial software projects, and investigate how sketches made with our tool are re-used and changed during projects. This will allow us to gather feedback about the quality of sketches from people who will actually have to re-use these artifacts.

References

[1] M. Cherubini, G. Venolia, R. DeLine, and A. J. Ko, "Let's go to the whiteboard: how and why software developers use drawings," in *Proc. CHI*. ACM, 2007, pp. 557–566.

[2] N. Mangano, A. Baker, M. Dempsey, E. Navarro, and A. van der Hoek, "Software design sketching with Calico," in *Proc. ASE*. IEEE/ACM, 2010, pp. 23–32.

[3] M. D. Gross and E. Y.-L. Do, "Ambiguous intentions: a paper-like interface for creative design," in *Proc. UIST*. ACM, 1996, pp. 183–192.

[4] B. Tversky, "What do sketches say about thinking?" in *Proc. Spring Symp. on sketch understanding*. AAAI, 2002.

[5] U. Dekel and J. D. Herbsleb, "Notation and representation in collaborative object-oriented design: An observational study," *SIGPLAN Not.*, vol. 42, no. 10, pp. 261–280, Oct. 2007.

[6] D. Wüest, N. Seyff, and M. Glinz, "FlexiSketch: A mobile sketching tool for software modeling," in *Proc. MobiCASE 2012*, D. Uhler et al. (Eds.), LNICST 110, Springer-Verlag, 2013, pp. 225–244.

[7] D. Wüest, N. Seyff, and M. Glinz, "Semi-automatic generation of metamodels from model sketches," in *Proc. ASE*. IEEE/ACM, 2013, pp. 664–669.

[8] D. Wüest, N. Seyff, and M. Glinz, "FLEXISKETCH TEAM: Collaborative sketching and notation creation on the fly," in *Proc. ICSE*. IEEE/ACM, 2015, pp. 685–688.

[9] N. Mangano, T. D. LaToza, M. Petre, and A. van der Hoek, "Supporting informal design with interactive whiteboards," in *Proc. CHI*. ACM, 2014, pp. 331–340.

[10] J. C. Tang, "Findings from observational studies of collaborative work," *Int. Journal of Man-Machine Studies*, vol. 34, no. 2, pp. 143–160, 1991.

[11] C. Gutwin and S. Greenberg, "Design for individuals, design for groups: Tradeoffs between power and workspace awareness," in *Proc. CSCW*. ACM, 1998, pp. 207–216.

[12] C. Gutwin and S. Greenberg, "The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces," in *Proc. WETICE*. IEEE, 2000, pp. 98–103.

[13] M. Haller, J. Leitner, T. Seifried, J. R. Wallace, S. D. Scott, C. Richter, P. Brandl, A. Gokcezade, and S. Hunter, "The NiCE Discussion Room: Integrating paper and digital media to support co-located group meetings," in *Proc. CHI*. ACM, 2010, pp. 609–618.

[14] S. Greenberg and D. Marwood, "Real time groupware as a distributed system: Concurrency control and its effect on the interface," in *Proc. CSCW*. ACM, 1994, pp. 207–217.

[15] D. Loksa, N. Mangano, T. D. LaToza, and A. van der Hoek, "Enabling a classroom design studio with a collaborative sketch design tool," in *Proc. ICSE*. IEEE, 2013, pp. 1073–1082.

[16] J. S. Lee, D. Tatar, and S. Harrison, "Micro-coordination: Because we did not already learn everything we need to know about working with others in kindergarten," in *Proc. CSCW*. ACM, 2012, pp. 1135–1144.

[17] P. C. Shih, D. H. Nguyen, S. H. Hirano, D. F. Redmiles, and G. R. Hayes, "GroupMind: Supporting idea generation through a collaborative mind-mapping tool," in *Proc. GROUP*. ACM, 2009, pp. 139–148.

[18] H. Ossher, B. John, M. Desmond, and R. Bellamy, "Are flexible modeling tools applicable to software design discussions?" IBM Tech. Rep. RC24949 (W1002-054), 2010.

[19] J. Walny, J. Haber, M. Dörk, J. Sillito, and M. S. T. Carpendale, "Follow that sketch: Lifecycles of diagrams and sketches in software development," in *Proc. VISSOFT*. IEEE, 2011, pp. 1–8.

[20] 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 *Proc. CHI*. ACM, 2010, pp. 75–84.

[21] M. Johansson and M. Arvola, "A case study of how user interface sketches, scenarios and computer prototypes structure stakeholder meetings," in *Proc. BCS-HCI*. British Computer Society, 2007, pp. 177–184.

[22] U. B. Sangiorgi, F. Beuvens, and J. Vanderdonckt, "User interface design by collaborative sketching," in *Proc. DIS*. ACM, 2012, pp. 378–387.

[23] Q. Chen, J. Grundy, and J. Hosking, "SUMLOW: early design-stage sketching of UML diagrams on an e-whiteboard." *Softw., Pract. Exper.*, vol. 38, no. 9, pp. 961–994, 2008.

[24] R. Van der Lugt, "Functions of sketching in design idea generation meetings," in *Proc. C&C*. ACM, 2002, pp. 72–79.

[25] L.-C. Lee and W.-J. Wei, "Comparison study on sketching behaviors with and without interactive pen displays," in *Proc. CSCWD 2007*, W. Shen et al. (Eds.), LNCS 5236, Springer-Verlag, 2008, pp. 567–574.

[26] A. Kleppe, *Software Language Engineering: Creating Domain-Specific Languages Using Metamodels*. Addison-Wesley Professional, 2008.

[27] H. Qattous, P. Gray, and R. Welland, "An empirical study of specification by example in a software engineering tool," in *Proc. ESEM*. ACM, 2010, pp. 16:1–16:10.

[28] J. Sánchez-Cuadrado, J. De Lara, and E. Guerra, "Bottom-up metamodelling: An interactive approach," in *Proc. MODELS*, R.B. France et al. (Eds.), LNCS 7590, Springer-Verlag, 2012, pp. 3–19.