Sketching Software in the Wild

David Socha
Computing and Software Systems
University of Washington, Bothell
Bothell, WA, USA
dsocha@uwb.edu

Josh Tenenberg
Institute of Technology
University of Washington, Tacoma
Tacoma, WA, USA
jtenenbg@uw.edu

Abstract—**This paper argues that understanding how professional software developers use diagrams and sketches in their work is an underexplored terrain. We illustrate this by summarizing a number of studies on sketching and diagramming across a variety of domains, and arguing for their limited generalizability. In order to develop further insight, we describe the design of a research project we are embarking upon and its grounding theoretical assumptions.**

Index Terms—**Diagramming, diagrams, sketching, distributed cognition, interaction analysis.**

I. Introduction

In a 2011 article named "Draw me a Picture" [1] Grady Booch questions whether diagrams play any meaningful role in software development. "*[W]hy don't [software] developers draw useful diagrams? For that matter, why don't they draw diagrams at all?*" In answer, he conjectures that "*Developers (for the most part) don't draw diagrams because diagrams rarely offer any fundamental value that advances the developers' essential work.*" Booch also notes that of the several hundred architectural descriptions that he has collected "*virtually all of them fall short in describing the system's real architecture.*" Booch thus believes that not only are diagrams rarely drawn (when they could be), but they neither fulfill a useful role in the development process when they are drawn nor are they accurate representations of the systems that they are meant to describe.

This seems a fairly damning claim coming from one of the fathers of Unified Modeling Language (UML). Yet in leading up to this conjecture, Booch describes how valuable diagrams can be when he is helping a dysfunctional project get a better understanding of their architecture. "*I'll go into a project and ask the team to draw me a picture of their system's architecture. [...] However, this picture is just the start of our software therapy session. Once I get the team members to start talking about what their picture means, the real architecture begins to be unveiled.*" That description implies that the team gets a great deal of value from the conversations and embodied activity around the diagram, which seems to contradict Booch's prior conjecture that diagrams rarely offer any fundamental value.

Which is the case? Are diagrams useful, or not? What essential roles do diagrams play in a professional software development team, if any? Do developers tend to not draw diagrams? If so, why? If they draw them, in what ways are they useful? How do they function with respect to the conversations and activities among members of a software development team?

Given the importance that our discipline ascribes to diagramming techniques like UML and the amount of work that has been devoted to tools that support these diagramming techniques, it seems important to understand whether and in what concrete ways diagramming and sketching contribute to professional software development. Although there are tentative answers emerging from extant research, this question remains underexplored.

While the existing research on sketching and diagramming in a variety of settings and disciplines provides some insight into the above questions, many of the insights are difficult to apply to address Booch's dilemma. This is because: *sketches* are too often the focus of investigation, not their use in sketching activity; for those studies in which sketching is a focus, too often it is the *individual in isolation* who is studied, not groups who are using the sketching to mediate their activity; in most of these studies, the individuals studied are *students,* not expert practitioners; there is very little research on *sketching activity within software development*; and finally, the empirical research is virtually always undertaken within a *contrived setting*, not in the workplace.

In this paper, we describe the design of a research study to uncover deeper insights into how groups of professional software developers create and use diagrams and diagramming in their authentic work, i.e. "in the wild". In particular, the highly contextual nature of work [2] means that answering questions about how groups of professional software developers create and use diagrams requires that we observe these groups undertaking authentic software development activity in their places of work. Furthermore, it requires the use of video analysis to look for and analyze the fine-grained nature of the social and material circumstances and artifacts that support individual and social cognition. In describing this study, we provide an overview of the theoretical frameworks that ground our methodological choices.

II. Background

There is a great deal of empirical research on the role of sketches, diagrams and visualizations and the process of their creation, across a range of disciplines. Sketches, diagrams and visualizations span different levels of formalism from a napkin sketch to a syntactically correct UML model. This paper focuses on the entire spectrum. In examining this prior research, we identified five areas that limited their generalizability to groups of software developers at work, which we discuss in turn.

Diagrams versus diagramming. Prior research has often focused on characteristics of the sketch, diagram, or visualization created (the noun), not on the sketching, diagramming, or visualizing activity (the verb). For instance, Brescianai et al. [3] developed a *collaborative dimensions framework* to characterize different aspects of sketches, diagrams, charts, and other "conceptual visualizations". Because of this focus on the artefact, not the process, the main research methods used are semi-structured interviews and surveys, sometimes stimulated by reference to specific artifacts [4].

A focus on the sketch embeds the assumption that the marks on paper, whiteboard, or computer that an individual makes reflect what the individual believes about the conceptual system that they are referencing, that these representations are in essence externalized "mental representations". Yet research on situated cognition [5, 6] indicates that the external representations say rather little about how an individual goes about thinking and acting, how he or she *uses* the representations, and the extent to which these representations reflect what has been internalized. When internalized ideas are externalized (such as in a map, a drawing, an utterance, a technical tool) they become part of the "surround" that the person and others use for subsequent activity, which, when combined with an individual's embodied perceptual system allows for changes to a person's internalizations. This enables "cognitive looping" [7] between self and world for carrying out many activities, and gives rise to iterated perceptual-cognitive loops that are not possible with purely (internal) mental representations.

The act of bringing thoughts into material form, such as expressing architectural designs in sketches and models, is itself constitutive of and essential to cognitive activity [8]. The interplay of sign, perception, and cognition over the entire episode of activity is what is most important, which is never considered if diagrams (and not *diagramming*) are the research focus. In other words, sketches might not be direct reflections of a person's internal representation of a system or situation, but rather the external scaffolding necessary for the individual to carry out their activity. Sketches can only be fully understood in relation to their actual use in goal-directed activity.

Individuals versus groups. There have been several empirical studies of artists and designers, where the researcher has sought to determine the role that sketches play in artistic and design activity (e.g. [9, 10]). These studies, focused on the interaction between an individual and externalized representations, have served as the foundation for our understanding of "cognitive looping" as described just above.

Yet, externalized representations are often central to the system of distributed activity toward achieving the joint goals of multiple people. For example, Hutchins [11] describes how a team of six navy midshipmen use a *nautical chart* for the navigation of large ocean going vessels within restricted waterways. Such external representations mediate joint activity because they "out of the mind" of any particular individual and they are perceivable. This allows them to serve as available resources that others can draw on as needed when engaged in joint activity. In addition, people use speech to make specific reference to such externalizations that are mutually known to be jointly perceived. This serves as a foundation for groups of people jointly engaged in a task to develop what Clark calls *common ground*, i.e. the mutual knowledge that people have about the setting, the task, and one another's state of knowledge [12]. Thus, research focused only on individual use of sketches will overlook many of the key functions that sketches play in mediating *joint* activity.

Students versus expert practitioners. Because of access by researchers, a large number of the empirical studies on sketches and sketching use students as the research subjects (e.g. [13, 14]). Yet the thinking and activity of students may differ dramatically from that of experts. Experts possess deep domain knowledge [15]. They additionally bring this knowledge to bear on problems in the field, self-monitor their work, and work with speed and dexterity [16]. Experts develop prodigious skill and sensitivity in choosing and working with materials and tools [17], and adapt these to changing contexts [18]. In a range of human endeavors, experts develop a sophisticated repertoire of strategies for managing the complexities of human interaction within social settings. Practicing professionals thus have considerable expertise that they can bring to their domains of activity. Yet much of this knowledge is tacit, embedded in the taken-for-granted practices that are enacted within the social and material context in which they work [19]. A focus on students will not provide insight into the subtle and complex ways in which experts generate and use diagrams and sketches in their ongoing activity.

Software versus other domains. A considerable amount of the empirical research on sketching and diagramming (including those referenced above) concerns the use of these externalizations by artists, architects, and other designers. And although software development as a domain of activity shares some characteristics with these other domains, it also has unique characteristics that might reduce the transferability of results from research studies in other domains. These special characteristics include not only that computer systems are designed human artifacts (thus, in the process of their being designed, they cannot simply be "read" or "measured" as can the bio-physical systems that natural scientists study). But in addition, the computation itself is invisible, the systems are immensely complex due to the combinatorics of discrete systems, the manufacturing and distribution costs are dramatically less than for physical products leading to different design constraints, software systems are highly conformable, and the systems exhibit behavior over time [20, 21]. Thus, studying sketching and diagramming in other domains is unlikely to characterize this activity within software development.

Contrived settings versus workplace. Two recent studies that address the limitations voiced earlier in this section both involve the sketching activity of groups of expert software developers [22, 23], [24]. These studies, however, involve activity within contrived settings, one taking place in a lab environment, and the other at an OOPSLA *DesignFest*. Yet from workplace studies in other domains in which practitioners have been studied in their working context, we know that practitioners make considerable use of the social and material resources that exist within the setting of activity [17, 18]. And that such resources are not simply found, but rather, are deliberately created, collected, appropriated, and organized by practitioners so that they are to hand as needed.

In one of the few studies that we found of the use of sketches and diagrams "in the wild", Roth describes the ways in which practicing scientists use graphs in their everyday settings [25]. What he underscores is that practitioners make sense of the diagrams that they use and create by virtue of the *embeddedness* of the graph in context, its interrelationship to the people, activity, and concreteness of the setting. This suggests that studying the sketching and diagramming activity of groups of software developers "in captivity" will provide little insight into the larger systems of meaning and activity of which the sketches and diagrams are a part.

To summarize, although there has been considerable research on sketching and diagramming using a variety of methods in a variety of domains, their generalizability to the activity of expert software developers in the workplace is limited. In the next section, we describe the design of a research study that we are initiating to address these limitations.

III. Research Design

Grounded in the literature above, particularly that of situated and distributed cognition, our research study is intended to provide insight into the following types of questions: Why and how do groups of software developers diagram and sketch? In what specific ways do the representations, speech, and accompanying gestures and body movements combine to create meaning-in-context that furthers the joint work of the participants? How do software developers establish common ground during the session?

Our primary method of data collection will be to make audio-visual recordings of sessions in which professional software developers undertake joint sketching and diagramming in the wild as they do their authentic work in their place of work. We will instrument a shared area at the identified organization in which such sessions naturally occur with two or more video cameras directed toward the medium that they are using (e.g. a whiteboard, table surface, or workstation). The sessions will be recorded by either the researchers or the participants. Having participants record their own data has the advantage of being non-intrusive, and places the locus of choice with the members of the organization about what data to share with the researchers. The use of video to study professionals in the workplace was pioneered at Xerox in the 1980's [26]. But it is only in the last decade, with the dramatic drop in cost of video recorders and the wide availability of free and inexpensive video analysis software, that the use of video has become widespread [27].

Our primary method of analyzing the data is *interaction analysis*, "an interdisciplinary method for the empirical investigation of the interaction of human beings with each other and with objects in their environment. ... Its roots lie in ethnography (especially participant observation), sociolinguistics, ethnomethodology, conversation analysis, kinesics, proxemics, and ethology" [28]. Because of its focus on tool-mediated interaction among individuals carrying out activity within a particular setting, interaction analysis requires the use of audio-visual recordings so as to allow for both replay and data sharing among multiple researchers. In addition, some of these sessions will be replayed to some or all of the participants so as to have their commentary on the session (which will also be recorded), thus serving as an additional layer of data to augment the session recordings. For a subset of the post-session interviews, we will interview the participants separately or in groups in order to gain insight into our research questions concerning how different individuals at the same session construe the signs and sign referents, as well as the overall meaning and experience of the session.

We will also rely upon an ethnography of the organization in order to understand the enclosing system of relationships, values, constraints, and practices of the participants in the whiteboarding activities. It will be informal, what Geertz characterizes as "deep hanging out" [29], and the data generated will include field notes, photographs, and audio and video recordings of interviews. This will help us to answer our research questions concerning the demographic, cultural, physical and organizational surrounds of the people carrying out the work. Without an understanding of these contextual factors, it will be difficult for us to make sense of the behavior that we are observing.

Our analysis as a whole will be iterative and incremental, starting as soon as data is collected. This will allow us to validate and deepen our emerging understandings by reflecting them back to the organizational actors whom we have observed and spoken with. It will also allow us to make alterations to our data collection protocol as we come to better understand the setting and actors. Finally, involving the actors in this process may increase the value that the actors and their organization accrue from their involvement with our research.

IV. Implications and Conclusion

A number of practices in software engineering have the status of "conventional wisdom," assumed to be useful across a range of contexts. Diagramming and sketching would seem to be such practices given the number of software tools, textbooks, and courses devoted to them. Yet Booch challenges these assumptions in claiming that software developers either do not draw diagrams or do not employ them in a useful way [1]. Surprisingly, there is little research that we can draw upon to adjudicate between Booch's claim and conventional wisdom. That which exists too often concerns students in a non-software discipline carrying out individual work in a laboratory.

In this paper, we described the design of a research study just underway to provide deeper insights into this underexplored terrain of professional software developers in the wild. Grounded in distributed cognition theory and using video recordings and interaction analysis, we will investigate the situated use of sketches and diagrams by expert software practitioners in their everyday activities in the workplace. In addition to providing new insights about how experienced professional software developers collaborate, the proposed work could help inform the design, choice, and use of collaboration techniques in general, development of software systems to support collaboration, and provide pedagogical insights into how to train our students to be more effective software developers. Are diagrams and diagramming as valuable as we tell our students they are?

ACKNOWLEDGEMENTS

This paper benefited from feedback from Marian Petre of Open University and Andy Ko of University of Washington, Seattle. The authors also acknowledge the Helena Riaboff Whiteley Center at the Friday Harbor Laboratories of the University of Washington for providing a quiet and conducive space in which some of this work was carried out. Finally, we thank our study participants who have been so kind to extend to us the level of trust that is critical for this type of research.

REFERENCES

[1] G. Booch, “Draw Me a Picture,” *IEEE Software*, vol. 28, no. 1, pp. 6–7, Jan. 2011.

[2] S. Greenberg, “Context as a Dynamic Construct,” *Human-Computer Interaction*, vol. 16, no. 2, pp. 257–268, Dec. 2001.

[3] S. Bresciani, A. F. Blackwell, and M. Eppler, “A Collaborative Dimensions Framework: Understanding the Mediating Role of Conceptual Visualizations in Collaborative Knowledge Work,” *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pp. 364–364, Jan. 2008.

[4] 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 - CHI ’07*, 2007, vol. 1, pp. 557–566.

[5] G. Salomon, Ed., *Distributed Cognitions: Psychological and Educational Considerations*. Cambridge University Press, 1993.

[6] P. Robbins and M. Aydede, Eds., *Cambridge Handbook of Situated Cognition*. Cambridge University Press, 2009.

[7] A. Clark, *Supersizing the Mind: Embodiment, Action, and Cognitive Extension*. Oxford University Press, 2008.

[8] A. Clark, “Natural-Born Cyborgs?,” *Lecture notes in computer science.*, no. 2117, pp. 17 – 24, 2001.

[9] C. Van Leeuwen, I. Verstijnen, and P. Hekkert, “Common unconscious dynamics underlie common conscious effects: a case study in the interactive nature of perception and creation,” in *Modeling Consciousness Across the Disciplines*, S. Jordan, Ed. University Press of America, 1999.

[10] G. Goldschmidt, “The dialectics of sketching,” *Creativity Research Journal*, vol. 4, no. 2, pp. 123 – 143, 1991.

[11] E. Hutchins, “Learning to Navigate,” in *Understanding Practice: Perspectives on Activity and Context*, S. Chaiklin and J. Lave, Eds. Cambridge University Press, 1993, pp. 35–63.

[12] H. H. Clark and S. E. Brennan, “Grounding in communication,” *Perspectives on socially shared cognition*, vol. 13, no. 1991, pp. 127–149, 1991.

[13] A. Menezes and B. Lawson, “How designers perceive sketches,” *Design Studies*, vol. 27, pp. 571–585, 2006.

[14] J. Zhang, “The nature of external representations in problem solving,” *Cognitive Science*, vol. 21, no. 2, pp. 179–217, 1997.

[15] M. Chi, R. Glaser, and M. Farr, Eds., *The nature of expertise*. Lawrence Erlbaum Associates, 1988.

[16] D. Feldon, “The Implications of Research on Expertise for Curriculum and Pedagogy,” *Educational Psychology Review*, vol. 19, pp. 91–110, 2007.

[17] D. A. Harper, *Working knowledge : skill and community in a small shop*. Berkley ; Oxford: University of California Press, 1992, 1987, pp. x, 224p.

[18] M. Rose, *The mind at work*. Viking, 2004.

[19] H. M. Collins, “Tacit Knowledge, Trust and the Q of Sapphire,” *Social Studies of Science*, vol. 31, no. 1, pp. 71–85, 2001.

[20] D. Socha and S. Walter, “Is Designing Software Different From Designing Other Things?,” *International Journal of Engineering Education*, vol. 22, no. 3, pp. 540–550, 2006.

[21] F. Brooks, “No silver bullet: essence and accident in software engineering,” in *Proceedings of the IFIP Tenth World Computing Conference*, 1986, pp. 1069–1076.

[22] A. Baker, A. van der Hoek, H. Ossher, and M. Petre, “Guest Editors’ Introduction: Studying Professional Software Design,” *IEEE Software*, vol. 29, no. 1, pp. 28–33, Jan. 2012.

[23] M. Petre, A. van der Hoek, and A. Baker, “Editorial,” *Design Studies*, vol. 31, no. 6, pp. 533–544, Nov. 2010.

[24] U. Dekel and J. Herbsleb, “Notation and Representation in Collaborative Object-Oriented Design: An Observational Study,” in *OOPSLA ’07: Proceedings of the 22nd annual ACM SIGPLAN conference on Object-oriented programming systems and applications*, 2007, pp. 261–280.

[25] W.-M. Roth, *Toward an Anthropology of Graphing: Semiotic and Activity-Theoretic Perspectives*. Kluwer Academic Publishers, 2003.

[26] M. Szymanski and J. Whalen, Eds., *Making Work Visible: Ethnographically Grounded Case Studies of Work Practice*. Cambridge University Press, 2011.

[27] C. Heath, *Video in qualitative research: analysing social interaction in everyday life*. Los Angeles: SAGE, 2010.

[28] B. B. Jordan and A. Henderson, “Interacton Analysis: Foundations and Practice,” *The Journal of the Learning Sciences*, vol. 4, no. 1, pp. 39–103, 1995.

[29] C. Geertz, “Deep Hanging Out,” *The New York Review of Books*, Oct-1998.