Sociomaterial-Design in Global Software Development

Pernille Bjørn
University of California, Irvine
IT University of Copenhagen
Intel Center for Social Computing
pbra@itu.dk

ABSTRACT

Sociomaterial-Design seeks to connect design practices and relational ontology by insisting that artefacts do not have pre-determined boundaries but instead only comes into being when enacted in practice. Sociomaterial-Design provides an analytical lens for unpacking complex collaborative practices for the purpose of identifying the design space for artefacts without pre-determined boundaries. This paper is a first step in exploring how we can follow the Sociomaterial-Design approach when we conduct CSCW studies and design for global software development practice. The paper focuses on the 'documentation practices' within global software development and explores documentation practice by examining six empirical cases from literature. Examining these cases, the sociomaterial practice of 'documentation' in global software development is unpacked as a multiplicity. The findings suggest that by taking a Sociomaterial-Design approach, 'documentation' in global software development emerges as a practice embedding the dynamic nature of 'documentation' as well a the relational practices by which 'documentation' is enacted over time and by different people. It points to how 'documentation' is a complex intertwined practice of creating and keeping up to date a diverse set of relations critical when executing large complex software development projects across geography and time. This finding extends the space for what it means to design technologies supporting documentation practices within global software development.

Categories and Subject Descriptors

K.4.3 [**Computer Supported Cooperative Work**]: Global software development

General Terms

Management, Design, Human Factors, Theory.

Keywords

Sociomateriality, design, CSCW, global software development, Sociomaterial-Design

1. INTRODUCTION

Sociomateriality is an epistemologically approach stipulating that the material properties of technology matters, however cannot be understood outside the entangled organizational practices by which is it embedded (Suchman 2007). Sociomateriality has its roots in the theoretical foundations closely related to the field of Computer Supported Cooperative Work (CSCW) and have recently received increased interest within areas such as information systems (Jones 2013; Mazmanian, Cohn et al. 2013) and organizational practices (Leonardi, Nardi et al. 2012). In this stream of literature fundamental CSCW principles, such as that the material physical properties of technologies matter in organizational practices, have been highlighted in terms of re-introducing the IT artifact back into research on organizational practices (Orlikowski 2007). The material properties have been at the center of attention for much CSCW research, and it is renewed in the recent raise of socially embedded technologies (Randall, Wulf et al. fortcoming). Thus, you might ask what is so special about sociomateriality and what can this approach bring to CSCW?

While organizational research have applied sociomateriality as a way to bring in the material properties of artifact, the theoretical approach offers a different contribution to CSCW, namely that while the material properties matters, we cannot understand these material matters outside the sociomaterial practices by which the artefacts are enacted. In this way, sociomateriality sets a new agenda to the practices of design, namely that the nature of technology can never be understood outside the sociomaterial practice in which is it embedded. Technological artifact within sociomaterial agenda do not have pre-determined boundaries, instead boundaries are made and remade – they are enacted in practice and of a temporal nature.

So what does it mean to design artefacts without pre-determined boundaries? How can we even think about artefacts where the physical properties do not comprise the boundaries? Sociomaterial-Design (Bjørn 2012) suggests that what makes the boundaries of artefacts is a particular practice, a temporal activity or a *doing*. When people enact artifact in sociomaterial practices, they engage in bounding practices, where they *bind-together* in a hyphened-structure while [bracketing-out] what becomes part of the entity, and what becomes outside the entity (Bjørn and Østerlund, in progress). The material properties are not what make the boundaries of e.g. a SCRUM board, as in the physically squared whiteboard hanging on the wall. Instead, the boundaries for what make the SCRUM board comprise the sociomaterial practices by which the artifact is enacted. There are uncountable numbers of bounding practices, which makes artefacts within organizational practices such as global software development, and we cannot expect to embrace the complete set of bounding, since there is no such thing as complete set of bounding practices.

What should we do as sociomaterial-designers, when we agree that all organizational practices are sociomaterial, which means that artifact do not have clear pre-determined boundaries? We cannot design sociomaterial practices, since these are emergent, but what we can do it to design artifact not knowing which bounding they will engage. If we are to design artifacts, which are to be enacted in the sociomaterial practices involved in global software development, what we can do is to pick apart the

Workshop on Global Software Development in a CSCW perspective, Computer Supported Cooperative Work 2014, February, Baltimore, United States.

sociomaterial entanglements, which makes the artefacts in current practices.

The question then becomes, which bounding practices are critical to explore, when we study global software development? How can we understand the enactment of artefacts in global software development in terms of bounding practices? Which artifact should we explore in all their sociomaterial practices in global software development? How do we study and design artefacts as sociomaterial entities?

These are some of the questions, which are critical in the exploration of the collaborative work involved in global software development. Such analyses can take many forms and perspectives, however essentially the starting point is to study how existing artefacts are enacted within the collaborative sociomaterial practices and use this as point of departure. In this position paper, we explore the bounding practices of one of the most important artefacts within global software development; namely 'documentation'. Now what *makes* the artifact of documentation is not an easy question, since documentation forms multiple practices by multiple people doing diverse activities creating multiple traces of different material properties. By examining empirical cases reported upon in the literature on global software development, this position paper explores the boundings, which make the artifact *documentation*, and zoom in on how documentation is enacted in practice. This exploration serves as the first step in identifying which bounding practices are particular important in global software development in terms designing collaborative artefacts for documentation, which again leads us to discover and propose how sociomaterial-designers can design for global software development.

2. DOCUMENTATION

At a first glance, documentation in software development appears as the collection of documents (reports, technical descriptions, source code, and notes etc.), which describe the IT application under construction from multiple perspectives. In this way, documentation initially appears as "text" in its most broad sense, and the boundaries for the documentation artifact might appear as text documents (digital or print-outs), which the software developers use in their work constructing the IT application. However, documentation is not just a collection of documents, it is also a practice; the practice of documenting. Documenting practices concerns reading/writing activities, where software developers create new or engage with existing documents, identifying relevant pieces of text for the task at hand, while revising existing text before these are put aside (Christensen and Bjørn 2014). Therefore, we need to unpack what documentation practice is and how this serves as the foundation for how documentation artefacts are enacted in practice.

Since the purpose here is to investigate document practices at a more general level, and not only as it appears within one empirical case, this analysis have examined documentation practices as it is described in six previous published papers on global software development (Herbsleb and Grinter 1999; Souza, Redmiles et al. 2003; Boden, Avram et al. 2009; Jensen and Bjørn 2012; Søderberg, Krishna et al. 2013; Matthiesen, Bjørn et al. 2014). Each paper presents empirical data on global software development, and even though documentation is not the center of attention in the papers, information about documentation practices can be retracted from the papers. Based upon this literature different documentation practices were identified and a multiplicity of sociomaterial practices of documentation emerged. Analyzing the empirical material, the strategy was to identify what in the paper concerned documentation practices, and then create overall categories for what documentation practices entails within global software development.

2.1 Documentation as Collaborative Practice

Documentation is a collaborative practice within global software development highly linked to other practices such as programming, testing, or refactoring – and is thus part of what makes the documentscape of global software development (Christensen and Bjørn 2014). Documentation practices are organized in different ways related to the scope and size of the software development project, as well as the complexity of the software application. Time spend on documentation might, within some methodologies (e.g. agile development), be reduced to a minimum (Boden, Avram et al. 2009), while in other be a fundamental activity of creating transparency in the processes and thus requiring many resources (Søderberg, Krishna et al. 2013).

Collaboration is when multiple participants are required to coordinate their individual activities in order to solve a mutual task (Schmidt and Bannon 1992). Collaboration thus requires extra work – articulation work (Strauss 1985; Strauss, Fagerhaugh et al. 1985). Documentation in software development is often viewed as 'extra work' (Boden, Avram et al. 2009; Matthiesen, Bjørn et al. 2014), which in particular is needed when the developers are geographically dispersed. However, if we view documentation practices as only articulation work and not part of the 'real work' of software developers we miss important aspects of documentation practices. Documentation practices are collaborative practices, which includes both work and articulation work. Work might comprise writing, reading, and revising documentation documents, while the articulation work might comprise the coordination and awareness practices by which the developers organize the documentation practices. Documentation practices includes the negotiation and communication of e.g. how to interpret the requirement specifications and turn these into source code (Jensen and Bjørn 2012). Documentation is thus a particular activity and practices within global software development, which serve our attention when we conduct research on global software development.

2.2 Documentation as bounding practices

While documentation practices on a high-level concerns the work and articulation work by which individual write, read, and revise documentation, we need to zoom closer into the practices to unpack the sociomaterial practices of documentation.

When we place documentation practices at the center of the analysis it becomes clear from all the empirical cases that a fundamental feature of documentation is to bind together diverse artefacts (as in source code, program environments, requirement specifications etc.) across the project and people. Bounding practices refer to the practices by which participants enact artefacts by engaging in mutual relations and linking together several artifact, movements, and locations (Bjørn 2012). The *bounding practices* of documentation creates sociomaterial entities such as [tester-developers-test-manager-documentation-multiple-locations] or [user-manuals-application-developer] (both examples from Souza, Redmiles et al. 2003). What these entities means is that the entity is a combined nature between people, artefacts, and activities, which are linked together in different ways at different times. What is interesting with this type of analysis is that focus becomes on the *connections* rather than on

singular elements. It is the relations between the elements, which makes the sociomaterial entity, not the elements by themselves. Also it emphasis the malleable nature of the sociomaterial entities, as these changes continuously. So when Herbsleb and Grinter (1999) in their detailed descriptions of software development practices point to the importance of coordinating refinements in source code as well as handling unpredictability by making it possible to find ways to return to original source code, and then at the same time describe how documentation tools were inadequate to supports such documentation practices, these empirical observations point to the multiplicity and complexity of documentation practices, which is difficult to turn into the design of useful tools.

By thinking about documentation as a sociomaterial practice, where people enact artifacts in diverse ways, through bounding practices connecting and disconnecting relations in a timely manner, the entities of what makes documentation changes. Documentation practices includes the work of identifying and linking information between problem reports (descriptions of bugs and how to reproduce them), which might be relevant for the documentation of the final application (Souza, Redmiles et al. 2003). Here the linking practice is the center of attention, and documentation practices become part of the work involved with bug reports. Documentation information might not always be captured in formal documents, but instead within informal specification refinements (Herbsleb and Grinter 1999), thus the bounding practices includes how multiple artifacts of different sorts are linked together in different ways depending upon the task at hand. Bounding practice in global software development not only concerns the participants directly involved with programing, it also includes the relations between the vendor and the client (Søderberg, Krishna et al. 2013), and as such the bounding practices also includes the practices by which documentation is part of creating transparency across the organizations.

Documentation practices are depending upon peoples' competences in identifying which artefacts and practices to link together, which require a different set of competences than programing skills (Matthiesen, Bjørn et al. 2014). Documentation practice is a practice by which contextual knowledge (Jensen and Bjørn 2012) about the domain, the application, the technical specifications, and the software developers are brought into play in an artful integration of bounding practices. Thus, when we are designing documentation tools, we need to think about documentation not as a stable individual activity, where 'texts' are created for others in the future to understand the technical application and use. Instead, the argument in this paper is that documentation is a collaborative practice, which is seamlessly integrated within other practices involved in software development (e.g. programing), by which individuals are dynamically connecting and disconnecting information, knowledge, and artefacts in diverse ways.

3. CONCLUSION

Creating useful tools and artifacts to support documentation practices is difficult, since documentation is a multiplicity of practices which requires different types of tools and artifact at different times and to be used by a diverse set of people. If tools are not able to handle the complexity of documentation practices there is a risk that these tools will be abandon (Herbsleb and Grinter 1999). Documentation tools tend to focus either on the overall descriptions of the technical system or on the changes and updates within the programming code, however the sociomaterial perspective suggest that we should not limit our perspective on documentation as either/or but instead as the multiplicity of diverse practices which are interlinked and dynamically changing. This points to a different type of departure when we are designing new collaborative technologies for documentation practices within global software development. Namely, the point of departure where the sociomaterial practices is at the center stipulating that artefacts do not have pre-determined boundaries, but instead are sociomaterial entities and should be treated as such. This paper is only the first step in this direction – and space did not make it possible to fully elaborate on the approach. However, hopefully it did provide some initial indications as to where such analysis could take us if we apply the Sociomaterial-Design perspective as analytical lens.

4. ACKNOWLEDGMENTS

This study is conducted within the context of the interdisciplinary and inter-organizational research project 'Next Generation Technologies and Processes for Global Software Development', a research project that is financially supported by the National Council for Strategic Research, Ministry of Science, Innovation and Higher Education in Denmark. For further information, see www.nexgsd.org.

5. REFERENCES

Bjørn, P. (2012). "Bounding Practice: How People Act in Sociomaterial Practices." Scandinavian Journal of Information Systems 24(2).

Boden, A., G. Avram, et al. (2009). Knowledge management in distributed software development teams: Does culture matter? International conference on Global Software Engineering (ICGSE). Limerick, Ireland, IEEE Press: 18-27.

Christensen, L. and P. Bjørn (2014). Documentscape: Intertextuallity, sequentiality and Autonomy at work. Computer Human Interaction (CHI). Toronto, ON, Canada, ACM.

Herbsleb, J. and R. Grinter (1999). Splitting the organization and integrating the code: Conway's law revisited. ICSE. Los Angeles, ACM: 85-95.

Jensen, R. E. and P. Bjørn (2012). Divergence and convergence in global software development: Cultural complextities as societal structures. COOP: Design of cooperative systems. France, Springer: 123-136.

Jones, M. (2013). "A matter of life and death: Exploring conceptualizations of sociomateriality in the context of critcal care." MIS Quarterly special issue on sociomateriality.

Leonardi, P., B. Nardi, et al. (2012). Materiality and Organizing: Social interaction in a technological world. Oxford Press.

Matthiesen, S., P. Bjørn, et al. (2014). "Figure Out How to Code with the Hands of Others": Recognizing Cultural Blind Spots in Global Software Development. Computer Supported Cooperative Work (CSCW). Baltimore, USA, ACM.

Mazmanian, M., M. Cohn, et al. (2013). "Dynamic reconfiguration in planetary exploration: A sociomaterial ethnography." MIS Quarterly Special issue on Sociomateriality.

Orlikowski, W. (2007). "Sociomaterial practices: Exploring technology at work." Organization studies 28(9): 1435-1448.

Randall, D., V. Wulf, et al. (fortcoming). Designing socially embedded technologies in the real world, Springer.

Schmidt, K. and L. Bannon (1992). "Taking CSCW Seriously: Supporting Articulation Work." Computer Supported Cooperative Work (CSCW): An International Journal 1(1-2): 7-40.

Søderberg, A.-M., S. Krishna, et al. (2013). "Global Software Development: Commitment, Trust and Cultural Sensitivity in Strategic Partnerships." International journal of Management.

Souza, C. d., D. Redmiles, et al. (2003). Breaking the code, moving between private and public work in collaborative software development. GROUP. Florida, USA, ACM: 105-114.

Strauss, A. (1985). "Work and the division of labor." The Sociological Quarterly 26(1): 1-19.

Strauss, A., S. Fagerhaugh, et al. (1985). Social Organization of Medical Work. Chicago and London, The University of Chicago Press.

Suchman, L. (2007). Human-machine reconfigurations: plans and situated actions, Cambridge University Press.