Ready-to-hand Information and Computer-mediated Activity: Challenges, Opportunities, and Methods

Anders I. Mørch

Department of Education, University of Oslo, Norway
a.i.morch@iped.uio.no

Abstract: The theme of the workshop, coping with information, participation and collaboration overload assumes that access to information, participation, and collaboration is somehow problematic and probably more for older than younger people. In this position paper I will explore and discuss the problem along one line of research I have been involved in, addressing one problem (information overload) and generating another (participation overload). Then, I will describe a case study in co-creation in the domain of customer engagement and discuss the methods we used for understanding participation and collaboration, using a mixed methods approach. Implications for design (i.e. technology; human organization) to address the problems will be suggested as points for discussion in the workshop.

Keywords: co-creation, collaborative learning, collaboration, computer-based critics, mutual development, pedagogical agents, reflection-in-action

1 Introduction

Collaboration software and social networking technologies (SNT) succeed not primarily because of their user interfaces (high usability), but based on how they attract a large number of users. They serve as platforms for social interaction and user generated content and are not merely tools. Once hooked on a platform (e.g. Facebook or LinkedIn) it may be hard to leave because you have invested considerable effort in generating content (i.e. pictures, comments, group participation, adding contacts, liking people, rating performances, and so on). The technology automates parts of this by creating your account, and providing suggestions for relevant information to add.

Most recently, analytic tools have been integrated with STSs, for data collection, analysis, visualization and overview, and recommendations. By analyzing your data with statistical methods and predictive modeling techniques, these systems can predict how you will act and recommend what to do (e.g. purchase a certain product; join a discussion group, contacts to add, improve performance in some area/domain).

Proc. of Third International Workshop on Cultures of Participation in the Digital Age - CoPDA 2015
Madrid (Spain), May 26th, 2015 (published at http://ceur-ws.org).

2 Coping with Overload

In a series of research efforts spanning several years, in different research groups (HCI, AI, CSCW, CSCL) at different institutions in US and Norway, I have been involved in solving problems, generating problems, and researching problems pertaining to information, participation, and collaboration overload. I summarize the outcome of these efforts, and raise some issues for further work and discussion.

2.1 Critiquing Systems Supporting Reflection-in-action

Critiquing systems (originating in the KBS/HCC group at CU Boulder) challenged intelligent tutoring systems in domains in which optimal solutions were not attainable (Fischer et al., 1991), most notable design. Whereas problem solving aims at optimal solutions, design is about alternatives (possible solutions) and re-combinations within a dynamically constrained space (moving target). Design is characterized by multiple solutions, some better than others, according to a set of subjective criteria (constraints) that include user (client) requirements, building codes, safety standards, argumentation, designers' preferences, and so on. Computer-based critics operates in this design space and act by informing users (e.g. novice designers) about what moves they can make to create better designs. By doing this, critics divide the design process in two sub processes: construction and argumentation. Construction is the activity of graphically creating the form of the solution by direct manipulation of graphical building blocks, and argumentation is the activity of reasoning about the problem and the possible solutions, e.g. considering what to do next, the pros and cons of the different alternatives, the consequences of making certain moves, and which course of action to chose (McCall, Fischer & Mørch, 1990).

This distinction led to the notion of integrated design environments, consisting of a domain-specific construction kit and a hypertext system for representing argumentation (Fischer et al., 1991). Computer-based critics create an "interruption" of the construction situation (like a human critic standing behind your shoulder and giving advice for how to improve a design sketch) when the spatial configuration of the building blocks constitute a "violation" of one or more of the design rules. The user interface of the integrated design environment was named Janus (after the Roman god of two faces in opposite direction), and theoretically it was inspired by Donald Schön's notion of Reflection-in-Action (Schön, 1983), which we interpreted to mean that general information for solving a design problem should be available and relevant at the time the information is needed, *thus coping with information overload*. Using today's terminology, Schön's theory suggested the integration of a web based information system (e.g. a discussion forum) with a domain oriented design environment for artifact creation, using automated analysis (analytics) to switch between two modes of designing (constructive design and argumentative design).

Analytic engines built into contemporary networked environments (e.g. e-commerce sites, social media, learning technologies) can generate overviews and recommendations based on statistical methods, predicting how a user might act in a new situation compared to how other users with a longer "forward trajectory" have acted in the past. Depending on the degree to which the "trajectories" of two users do align, this approach will (or will not) solve the information overload problem.

2.2 Pedagogical Agents Prompting Participation and Advising Collaborative Inquiry in a Distributed Collaborative Learning Environment

At the University of Bergen we further developed the idea of critiquing for application to collaborative learning environments, calling the critics for pedagogical agents (Jondahl & Mørch, 2002). As with critics, the pedagogical agents had "rules" for modeling domain knowledge, and these rules represented what we knew about participation in an inquiry based discussion forum called Future Learning Environment (FLE) (Dolonen, Chen & Mørch, 2003). We found this "domain" harder to understand than the domain of kitchen design implemented in the Janus system, and consequently the rules by which we programmed the pedagogical agents more speculative and likely to be modified and refined (Mørch, Dolonen & Nævdal, 2006). Two of the rules in pseudo code form are: *if less activity than the average participant (in number of postings), then suggest higher activity; if there are many unaddressed problems or questions in the forum, then suggest addressing one of them by an answer or hypothesis*. In this way we not only addressed the information overload problem (inherited from critiquing systems), but also *generated a new problem* (participation overload).

2.3 Mutual Development and Co-creation

Mutual development (Andersen & Mørch, 2009; Mørch & Andersen, 2010) is a technique for co-creation of software artifacts through collaboration by two groups of stakeholders: professional software developers and end-user developers. End-user developers create local adaptations of a software product for personal or organizational needs, and professional developers create new versions of the software for sustaining their practice and increase revenue for their business. These end-user developers have much in common with "lead users." A lead user (von Hippel, 2005) is an early adopter of a new innovation, or someone who likes to experiment with the use of an existing product, or someone who creates an adaptation to a product based on knowledge of a related product. Product developers will often seek out lead users for feedback on early (beta) releases before they hit the market.

For a professional organization (e.g. a software house) to incorporate a new feature first proposed by an end-user (e.g. a customer) into an existing line of products would normally require multiple levels of collaboration. Developers collaborate when they provide tools for communication and information sharing with end-user developers and when they accept end user proposals for features in new releases (Andersen & Mørch, 2013). Incentives are needed to make collaboration work; and contracts may be necessary in order to handle ownership of a new innovation.

Mutual development involves multiple stakeholders, often in asymmetrical (e.g. user-developer) relations. It starts with communication for the purpose of building a common understanding. It continues by improvement request proposals or hacks submitted by end user developers, through collaboration with other end user developers and with professional developers, the latter selecting and filtering out good proposals for further work and incorporation. Research methods to identify and study these phenomena (i.e. communication to build common understanding and multi-disciplinary collaboration) benefit a mixed methods approach (combining qualitative and quantitative research methods) (Fugelli, Lahn & Mørch, 2013), especially when the user population is large, such as in crowd sourcing and mass collaboration (Tapscott & Williams, 2007).

This brief presentation summarizes the work my colleagues and I have been involved in over a number of years in a series of efforts in system building (software applications) and empirical studies in user organizations, using techniques from HCI, AI, CSCW, CSCW and EUD to cope with information, participation, and collaboration overload, which can be summarized as problem framing experiments:

1. *Solving a problem*: information overload with computer-based critics, finding information relevant to the task at hand;
2. *Creating a problem*: participation overload with pedagogical agents by asking users to increase their participation in an online collaborative learning environment in terms of quantity (number of postings) and in terms of quality (choosing the appropriate inquiry type for a new posting);
3. *Studying a problem*: research methods using a mixed methods approach for understanding communication and collaboration practices in a co-creation community in customer-initiated software product development.

3 Challenges, Opportunities, and Methods for End-user Development

There are multiple ways to address the aforementioned problems. EUD has a role to play. Here are some open issues for further work and discussion at the workshop:

- EUD in design; e.g. components and rules of composition; how to make design environments modifiable in terms of composition (design) and recombination (redesign)?
- Ill-defined problems and EUD, i.e. refinement of rules to model a domain that are crudely rendered at beginning;
- Rules for combining software components (tools); rules for combining learning resources; rules of participation; rules of interaction; rules of collaboration,
- Constraints in EUD-enabled SNTs and collaboration software: in terms of roles, interaction patterns, social structures, etc.
- Learning analytics and EUD; what should be the role of EUD in LA research
- Research methods: What are "appropriate" methods for analyzing problem situations, identifying alternative (possible) solutions, combinatorial limitations, etc.
- How to analyze EUD activity and visualize the activity to 1) end user developers, 2) professional developers, and 3) other stakeholders (e.g. customers, managers)?

References

1. Andersen, R., and Mørch, A. I. (2009). Mutual Development: A Case Study in Customer-Initiated Software Product Development. In V. Pipek, M.B. Rosson, B de Ruyter and V. Wulf (Eds.). *Proceedings 2nd Int'l Symposium on End User Development (IS-EUD 2009).* Berlin Heidelberg: Springer, pp. 31-49.
2. Andersen, R. and Mørch, A.I. (2013). Get Satisfaction: Customer Engagement in Collaborative Software Development. In *Proceedings of the 4th International Symposium on End-User Development*, Springer-Verlag, Berlin, Heidelberg, 235-240.
3. Dolonen, J., Chen, W. and Mørch, A. (2003). Integrating Software Agents with FLE3. *Proceedings of CSCL 2003.* Kluwer Academic, pp. 157-161.
4. Fischer, G., Lemke, A.C., McCall, R. and Morch, A.I. (1991). Making Argumentation Serve Design. *Human-Computer Interaction*, 6(3&4), pp. 393-419.
5. Fischer, G., Lemke, A.C., Mastaglio, T.W., and Morch, A.I. (1991). The Role of Critiquing in Cooperative Problem Solving. *ACM Transactions of Information Systems*, 9(2), pp.123-151.
6. Fugelli, P., Lahn, L.C. and Mørch, A.I. (2013). Shared prolepsis and intersubjectivity in open source development: Expansive grounding in distributed work. In *Proceedings of the 2013 conference on Computer supported cooperative* work *(CSCW`13).* ACM, New York, NY, USA, 129-144.
7. Jondahl S, Mørch A. Simulating pedagogical agents in a virtual learning environment. In: Stahl, G, ed. *Proceedings Computer Support for Collaborative Learning (CSCL 2002).* Boulder, CO, USA: Lawrence Erlbaum, 2002:531–532.
8. McCall, R., Fischer, G., and Mørch, A. (1990), Supporting Reflection-in-Action in the Janus Design Environment. In *The Electronic Design Studio: Architectural Knowledge and Media in the Computer Era*, 247-259. CAAD Futures. Cambridge, Massachusetts: The MIT Press, 1990.
9. Mørch, A.I. and Andersen, R. (2010). Mutual Development: The Software Engineering Context of End-User Development. *Journal of Organizational and End User Computing*, 22(2), pp. 36-57.
10. Mørch, A.I., Dolonen, A.I. and Nævdal, J.E. (2006). An Evolutionary Approach to Prototyping Pedagogical Agents: From Simulation to Integrated System. *Journal of Network and Computer Applications*, 29(2-3), 177-199.
11. Schön, D.A. (1983). The Reflective Practitioner: How Professionals Think in Action. New York, NY: Basic Books.
12. Tapscott, D. and Williams, A.D. (2008). *Wikinomics: How mass collaboration changes everything*. London, UK: Atlantic Group.
13. Von Hippel, E. (2005). *Democratizing innovation*. Cambridge, MA: MIT press.