Socio-Informatics: Intertwining Analytical and Design-oriented Research into Social Practices

in: V. Wulf; V. Pipek; D. Randall, M. Rohde; K. Schmidt; G. Stevens (eds): Socio Informatics – A Practice-based Perspective on the Design and Use of IT Artefacts, Oxford University Press, Oxford 2018, pp. 541-550

The various investigations in this book can be characterised as putting forward an argument for a non-deterministic relationship between investigation and design for social practices in which IT artefacts have an acknowledged relevance or a potential. The term 'practice', it is clear, provides a focus for careful analysis of the ways in which people orient to the use of technology over time, the way in which technology mediates behaviours and the various constraints, technological and otherwise, under which people operate. Nevertheless, it has been argued throughout this book that very little purchase on 'practice' is provided by naïve recourse to sociological theorising. To varying degrees, all the authors in this book recognise the need for concept development which specifically orients to the problem of design and development. This has never been more relevant, given the rapid technological changes evident almost everywhere we look.

The Computer Supported Cooperative Work (CSCW) research community, in particular, has long sought to examine the relationship between technology and 'work' in relation to the support one can provide for the other. Such a position is in sharp contrast with approaches which aim to automate work, in the sense of fully replacing human activities by machines, e.g. in the traditional mainstream of Artificial Intelligence (AI) community. However, supporting, in our sense, typically means engaging with and often accepting these practices may change– at least on a micro level. As part of a practice-based design agenda, the term 'support' needs, then, to be developed more carefully, since it may well include the formalization and even the automation of certain elements of human activities (Christiansen 2016). The critical issue here, as always, is to understand more completely just how change might ramify rather than crossing one's fingers, hoping for the best and ignoring possible ramifications.

This volume brings together contributions from authors with different foci, but with a passion for a shared endeavour to come to more accountable, transparent, empowering, and sustainable ways to foster technology-related practice improvements. This epilogue aims at setting out an agenda or at least a number of questions and milestones that illustrate potential pathways in our effort while considering technological trends as well as current and future working conditions of practitioners and researchers. We would like to think of an agenda of 'practice-based computing' as forming a discourse on what we would call 'socio-informatics', although that term has been widely used to mean a number of different things.

The term 'socio-informatics' was first used by some of the editors around the year 2000. Having worked during the 90s at a traditional computer science department, they came to the conclusion that applied computing suffers from some rather fundamental incoherencies and deficits, deeply ingrained in the historically grown identity of the field (see Rohde and Wulf 2011; Schmidt 2011). They included:

- limitations with regard to research methods, notably focusing on 'top down' approaches to design, and process models which emphasised the 'formal' character of the organization. In sum, focusing on the technical in the socio-technical equation,

- limited understanding of its theoretical foundations, mainly focusing on mathematical or formal theorems,

- problematic epistemological positioning, judging the validity of insights independently of the (social) context of its application.

CSCW, at its inception, and specifically in its European instantiation, aimed to identify a new research paradigm, one which might overcome the given deficiencies and (re)introduce some notion of the 'social' in computing. Our point, however, is that we are a long way from producing a coherent alternative. While without question rooted within a CSCW community, many practitioners have identified a variety of interests which may require new ways of thinking about the investigation/design relationship, for instance when examining 'non-work' environments like the home, games or IT Systems with a general role in society (e.g. technology affording an 'electronic' democracy), or when trying to understand how principles such as participatory design might be worked out in practice in these new conditions. Not least, making sense of more long term shifts in the way IT systems were embedded and used in different contexts seemed to be a complex and unresolved problem. Information systems (IS) discourses around notions of 'socio-technical systems' can seem both arbitrary and argued at such a rarified theoretical level as to be useless for practical design and development work. Finally, the technology landscape underwent revisions at a faster pace than theory-led researchers could work. These deficits have been alleviated by a number of approaches, some of which are discussed in this book. It remains the case, however, that making sense of 'practice' and its ramifications across settings and domains is a radically unfinished project.

Balancing investigation and design work remains, we would argue, a central objective of CSCW and cognate disciplines. 'Informatics', then (which is used in German synonymously for 'Computer Science', but also with a connotation of applied mathematics and information processing sciences), remains a useful way of reminding us that design is still an integral feature whilst remaining commited to ways of overcoming the limitations of the above. In our understanding, socio-informatics is still a field of informatics – it remains an ambition to build innovative IT artefacts and to inform the work of programmers and others involved in technology development processes. Investigation into practice, however, may entail investigations into a gamut of interests, including those of (sometimes ill-defined) users (as 'customers' of technology development) and of the designers (as 'manufacturers' of IT artefacts to be developed), and, of course, ourselves as researchers trying to understand and improve the landscape defining the relation between technology production and technology use. Moreover, we felt the need to establish a language and conceptualizations that did not prioritize one practice as given (ICT development) while other practices are under reconsideration (practices of potential ICT 'users'). We have outlined a number of concepts and approaches in this book to foster (self-)reflection among different practitioners who are concerned with or affected by new technological developments.

Some part of the objectives rehearsed above can be seen as a reaction to an epistemological monoculture, one where science and engineering models, or the logic of process modelling, or the presumed efficiencies of new programming techniques all act in roughly the same direction. A 'practice' perspective, it can be argued, entails a certain methodological and normative pluralism, such that careful comparison of cases, where similarity and difference are acknowledged, at a level appropriate to the different interests in play, might result in a genuine transferability.

In this volume we aimed to give a stronger foundation to these practice-oriented approaches around IT development. We can distinguish two rather different accounts of 'practice' here. (1) a concept-historical one developed by K. Schmidt in this volume and (2) practice-theoretical ones, outlined by, inter alia, A. Reckwitz. Schmidt provides an analysis of the twisted history of the concept of 'practices', spanning

over the last 2000 years. When it comes to the account's relevancy for IT design, he points to the importance of the interplay between rules, principles, and situated activities – a topos already deeply engrained in the CSCW discourse. Reckwitz (2002) develops his 'practice' term out of a meta-analysis of current sociological theorizing – a pattern of academic work which Schmidt in a second contribution to this book harshly criticizes. It is evident that the various contributors to this book do not always agree on what might be entailed in the characterization and use of the 'practice' term. Some have found that Reckwitz's meta-analysis offered some stimulation to their design work (see Wulf et al 2011 and 2015). These distinctive accounts, whatever position one takes, are both sensitive towards the developmental trajectories of social practices due to the introduction and design of IT artefacts. From a design point of view, however, in and of themselves, they lack a strong empirical foundation - an argument Schmidt puts forwards when criticizing the practice theorists. The issue, then, has to do with the role of theory, and what kind of theory is likely to prove most useful for the purposes we designate. As indicated repeatedly in this book, useful theories must orient to both analytic investigation and design, and need to do so while recognizing the longer term transformations that occur as new technology is designed, introduced, used and, in some cases, appropriated. They (theories) can, of course, be used in a variety of ways, including to determine what kind of analytic work should be done (a role we reject) or in that 'sensitising' or 'illuminating' way that Herbert Blumer famously recommended. At this point, we believe that Randall et al. (2007) still offer a very decent and yet flexible set of suggestions as to how we might go about the empirical investigations implied.

A design-oriented account needs, then, to provide the detailed, contextual descriptions that the 'practice turn' has always implied, but needs as well to engage with potentialities. Analytical accounts of existing or potential practices are not sufficient; they have a target they aim to inform and engage: technologically knowledgeable designers on the one hand and, on the other hand, (potential) users who find enough reason and motivation to reconsider their practices based on a recognition of possible improvements and perhaps become actively involved in re-design processes. As researchers in this arena, on the given assumption that we are engaged in both investigation and design, we are required to reflect upon our own practices as well, not least because at some point we will be involved in decisions about how, and to what extent, we can legitimately generalise from our findings and, just as much, what elements of the designs that result can also be re-used, form part of modular constructions, or support end user development.

Design case studies, recommended by the Siegen School in this book, can be understood in the tradition of a (participatory) action research paradigm (Whyte 1991; Hayes in this volume) using technological interventions to understand the dynamics of evolving practices in the domain at hand. In doing so, they lead to the design of innovative IT artefacts and accounts on their long-term appropriation, going at least some way to providing guidelines for the conceptual and building work that forms the core of our discipline. Aside from an intensified reflection on the practices involved in this kind of endeavour, the methodological foundations of IT development need to be constantly challenged. Where classic accounts of these methods in computer science focus on the completed product (hardware, software or both), we can choose to look at a (successfully) established usage as the anchor point in technology development. Stevens and Pipek (in this volume), in this way, reflect on technology appropriation studies and their consequences for design, also discussing a potential theoretical framework based on Bowker and Star's (2002) notion of Infrastructuring. 'Grounded Design' is the umbrella term used here to organize the generative effort of developing IT artefacts together with technologically enhanced practices, aligned with the idea of 'PraxLabs' and other flexible approaches to the issue of 'method', aimed at sustaining connections between development and use practices beyond traditional project-based structures. The focus is essentially pragmatic, rejecting any deterministic view of appropriate methodology in favour of general analytic commitments, reflected on in separate chapters: Randall's on

'Ethnography and Design', Wagner's on 'Participatory Design' and Hayes on 'Action Research', but all in varying degrees reflecting on the kind of involvement that one might expect and encourage from users.

To this end, the notion of End-User Development is considered as an arm of this kind of thinking. The aim here is to provide flexibility over longer periods of use and across several use situations, to support technology management by actors whose primary goal is not building the application at hand. EUD typically provides a 'gentle slope of complexity' that allows users to make an active choice on how far they want to immerse themselves in technological details of their infrastructure (Liebermann et al. 2006; Wulf et al 2008). Stevens and Pipek's chapter usefully elaborates on the social and collaborative notion of EUD.

The final chapter on meta research aims to elaborate on means to foster self-reflectivity among researchers, and its consequences for a research discourse in general (including the issue of comparability of results), and on the feasibility of methodological approaches for consideration of the possibility of practice improvement.

Socio-Informatics: Establishing a research discourse for practice-based computing

The practice-oriented perspective, it has been argued above, can foster a closer relation between development and use practices – and has the merit of providing the flexibility required for understanding new technological affordances (Ubiquitous Computing, Internet of Things) in a rapidly changing social and organizational context. Continued work on such practice-oriented perspectives could entail:

Technological Challenges

The technological challenges emerging from a practice-based research paradigm span from better support of appropriation activities to fundamental issues of software architecture and development:

- Appropriation infrastructures: Pipek (2005) suggested a new direction of technological development in supporting appropriation activities, specifically from the perspective of user-user co-operations. Stevens (2010) extended this line of thinking by suggesting appropriation infrastructures are to be understood as communication channels via which practioners of the technology development domain can interact with those from the use domain. The technological connection between these practice spheres still needs to be further explored as new technological ecosystems emerge (see Draxler and Stevens (2011)) for a related study into the Eclipse ecosystem).
- Designing Sociable Technologies: Sociable Technologies (Ludwig et al. 2014) describe an important extension of software artefacts being equipped with appropriation infrastructures in addressing the emerging world of physical artefacts in the realm of phenomena such as Ubiquitous Computing or the Internet of Things. Sociable technologies suggest the design of physical (and software) artefacts in ways that allow and actively support the sharing of use stories and configurations, and suggest to integrate additional sensors, actuators and communication channels that do not (only) relate to the functional purpose of a technology, but to a discourse on how to use it.
- Supporting communities of developers: To make practice-based computing more effective and efficient, there should be some reuse of components of newly developed and given IT artefacts. Therefore, it would make sense to have an open source repository, maybe connected to the documentation of the individual design case studies. That way a community of practice-based designers could draw not only on each other's design ideas but also on each other's code. Such

elements of code sharing would be obviously also beneficial in implementing appropriation infrastructures or sociable technologies.

- IT-architectures: An approach to reuse, can be supported by appropriately designed IT architectures. The design of IT architectures could take the results of different design case studies into account to compare the differences in requirements. Based on such a comparative analysis of design case studies, the software architects could decide for a module structure which would make it easily to change the more differentiated aspects while ingraining other architectonical decisions more deeply.
- End User Development: We have suggested that applications should be built in a way that end users, typically not being professional software developers, can modify these applications according to their changing needs (Lieberman et al 2006, Spahn et al, in this volume). This is important in supporting the appropriation of IT artefacts because with changing practices the IT artefacts may need to be tailored accordingly.

Methodological Challenges

Challenges when conducting individual design case studies have already been extensively discussed in this volume. In the following, we want to focus on infrastructural and methodological challenges resulting from a comparative approach to deal with a set of individual design case studies.

- Networks of Praxlabs: PraxLabs have been introduced in this book (chapter 10) as a research and development infrastructure that allows connecting to practice domains beyond the scope of a single design case study or project. Taking this approach seriously, we envision building a network of PraxLabs that covers various domains. The connection of neighbouring domains can not only become a resource for innovating practices and an ecology to allow technology manufacturers to develop their products but also a research environment to look comparatively at the frictions of similar or competing practices.
- Understanding appropriation: We already have a large variety of methods that structure the work of developing IT artefacts. These methods reach out into the practice domains that a IT development aims to support. However, there is not any complementary conceptualization of the creativity which happens in these practice domains when IT artefacts get appropriated. Following Donald Schöns' ground breaking work on the 'reflective practitioner', one could imagine to try understanding the IT users' craftsmanship on a level of theoretical reflection that IT design methodologies have reached.
- Comparative approach: We have still little experiences with buttom-up concept building when comparing different design case studies. Wulf et al (2015) and Betz and Wulf (in this volume) make first attempts in this direction. However, we need to better explore how to establish creative environment in which researchers with backgrounds in different case studies can be supported in finding conceptual similarities. We also need to explore how sensitizing concepts need to be framed and described to become meaningful in different contexts.
- Portefolio of Design Case Studies: To support the building of design related concepts and to make the insights manifested in individual concept better accessible over time, we need to learn how to build documentations of individual cases. Working on each case, typically creates a multitude of materials which are rarely preserved and shared beyond the individual research project.
- Metaresearch: An agenda for research into Grounded Design research practices is still to be elaborated upon. So far, we have looked into the emergence of new design ideas and the cooperation within research consortia and research groups (Dachtera et al 2014, Wan et al 2016, Randall et al. in this volume). We still need to better understand the particularity of the relevant design practices to allow other actors to draw on these experiences and insights. It is also an

issue to decide how far metaresearch might become embedded in the institutional environments of design research.

Epistemological Positioning

It remains to be difficult to foster an epistemological research position that emancipates itself from the origins of our societies' traditional conception of 'science' as aiming at deriving factual knowledge and universally applicable operational descriptions of observable phenomena. In fact, the speed of development that came with phenomena of 'digitalisation' throughout all spheres of modern lives makes changes tangible during one lifetime that took centuries before, combined with the potentials of 'digitalisation' to transcend and connect these spheres. These developments make a buttom-up approach to concept building and a stronger integration of reflective spheres not only possible, but necessary. In this sense this volume has tried to formulate a post-objectivist research paradigm in the domain of human centred computing.

We discussed Grounded Design, Design Case Studies and e-Portfolio ideas in this book to generate an integrated perspective on the development of practices as well as IT artefacts, to be complemented by a level of self-reflectivity in design research. We started a design-oriented discourse on the concepts of 'appropriation' and 'infrastructuring' (Pipek and Wulf 2009, Stevens et al 2010; Stevens and Pipek, in this volume) to develop a new terminology to describe these phenomena in a more accurate and integrated way. There are alternative conceptualizations (e.g. Engeström's Developmental Work Research 1988) which aim to address a similar level of integration (but were not able to foresee the phenomena and opportunities which 'digitalisation' provides today), and there may be additional ideas to come.

We point out that the arenas sketched above along a scale from the local and immediate to the global and reflective are a significant extension to traditional discourses in computer science and information systems. In their claim to universality, these discourses resemble approaches to system theory and cybernetics that ran through these disciplines in the 50s and 60s of the last century. Socio-Informatics remains concerned with the concrete spheres of practice improvement, technology innovation, and scientific reflection when discussing its rigor. Socio-Informatics aims to provide a roadmap for the investigation/design problematic, signposted by our running theme, the appeal to practice-based design.

References

Bowker, Geoffrey C.; and Susan Leigh Star (1999). Sorting Things Out: Classification and Its Consequences. Cambridge, Mass.: MIT Press.

Christiansen, L. R.; Harper, R. (2016): The Many Faces of Computational Artifacts. In Proceedings of COOP 16: 12th International Conference on the Design of Cooperative Systems, Trento, Italy, May 22th – May 27th 2016, Springer

Dachtera, J., Randall, D., Wulf, V.. 2014. Research on research: design research at the margins: academia, industry and end-users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14).* ACM, New York, NY, USA, 713-722.

Draxler, S.; Stevens, G. 2011. Supporting the collaborative appropriation of an open software ecosystem, in: *Computer Supported Cooperative Work (CSCW)* 20 (4–5): 403–48.

Engeström's Developmental Work Research 1988

Lieberman, Henry, Pateró, Fabio; Wulf, Volker (eds): End User Development, Springer, London 2006

Ludwig, T., Stickel, O., Boden, A., Pipek, V. (2014): Towards Sociable Technologies: An Empirical Study on Designing Appropriation Infrastructures for 3D Printing, in: Proceedings of the Designing Interactive Systems Conference (DIS 2014), At Vancouver, Canada, ACM Press New York

Pipek, Volkmar. 2005. *From tailoring to appropriation support: Negotiating groupware usage*. Oulun yliopisto.

Pipek, Volker; Wulf, Volker (2009): Infrastructuring: Towards an Integrated Perspective on the Design and Use of Information Technology, in: Journal of the Association of Information System (JAIS), Vol. 10, Issue 5, May 2009, 306-332

Reckwitz, A. (2002). Toward a Theory of Social Practices: A Development in Culturalist Theorizing. *European Journal of Social Theory.* May 2002, vol. 5, no. 2, 243-263.

Rohde, Markus und Wulf, Volker (2011): Aktuelles Schlagwort: Sozio-Informatik. Informatik-Spektrum, Vol. 34, No. 2, 2011, 210-213.

Randall, D., Rouncefield, M. and Harper, R. (2007). Fieldwork in Design. Berlin, Springer

Schmidt, Kjeld (2011). Cooperative Work and Coordinative Practices: Contributions to the Conceptual Foundations of Computer-Supported Cooperative Work (CSCW). London: Springer.

Stevens, G. (2010): *Understanding and designing appropriation infrastructures: artifacts as boundary objects in the continuous software development.* PhD Thesis, Faculty of Economic Disciplines, Universität Siegen, 2010

Stevens, Gunnar, Pipek, Volkmar, Wulf, Volker: Appropriation Infrastructure: Mediating Appropriation and Production Work, in: Journal of Organizational and End User Computing (JOEUC), Vol. 22, Issue 2, 2010, 58-81

Wan, L. Müller, C., Randall, D., Wulf, V. : Design of A GPS Monitoring System for Dementia Care and its Challenges in Academia-Industry Project, in: ACM Transactions on Computer Human Interaction (ToCHI), Vol. 23, No. 5, 2016, Article No. 31

Whyte, W.F. (ed.) (1991) Participatory Action Research. Sage, CA.

Wulf, V.; Pipek, V.; Won, M. (2008): Component-based Tailorability: Towards Highly Flexible Software Applications, in: International Journal on Human-Computer Studies (IJHCS), Vol. 66, No. 1, 1 – 22

Wulf, V., Rohde, M., Pipek, V., and Stevens, G. Engaging with practices: design case studies as a research framework in CSCW. *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, ACM (2011), 505–512.

Wulf, Volker, Claudia Müller, Volkmar Pipek, David Randall, Markus Rohde, and Gunnar Stevens (2015): Practice-based Computing: Empirically-grounded Conceptualizations derived from Design Case Studies, in: Wulf, V.; Schmidt, K.; Randall, D. (eds): Designing Socially Embedded Technologies in the Real-World, Springer, London, 2015, 111-150.