

The Locales Framework: making social thinking accessible for software practitioners?¹

Geraldine Fitzpatrick

Introduction

I was involved in a research group¹ that was building a collaboration environment called wOrlds (Fitzpatrick, 1995) to support people working together across space and time. The wOrlds system was supposed to overcome some of the problems with previous workflow-type approaches that had led one user to comment "Your system has been designed to be idiotproof, the trouble is we're not idiots" (Wastell, 1994, p.35). Given Suchman's description of the situated nature of work and the role of plans in work (1987), it was easy to understand this rejection.

In the wOrlds system, we tried to account for this social thinking about the situated contingent nature of work practices by providing shared spaces where work could happen dynamically, rather than *a priori* workflow prescriptions. When it came to a prototype implementation with a group of systems engineers however, it quickly became obvious that the wOrlds system would also likely be rejected but for different reasons - we had adopted too much of the metaphor of the spaces in which they worked instead of understanding how people used the spaces to get their work done (Fitzpatrick, 1996). Even though we claimed to be motivated by considerations of the social realm of work, we still

¹ Publisher's Draft: To appear in *Social Thinking, Software Practice*, Floyd, Dittrich & Klischewski (Eds), MIT Press

'got it wrong'. In getting it wrong, we also learnt more about the problem of designing interactive systems.

This chapter is a reflection on the next phase in our thinking about how to incorporate social thinking into our own software practices for collaborative systems design. The focus of this next phase was the development of the Locales Framework as a set of concepts based on a notion of *place* as the primary unit of analysis and design. It was conjectured that the framework could be used as a shared abstraction to mediate both understanding the nature of work and designing systems to support this work (Fitzpatrick, 1998).

The main point of this discussion is not so much to expound the details of the Locales Framework but to characterize the nature of one particular endeavor to build socially-situated software, and to reflect on the Locales Framework as an instance of the iteration in learning more about the problem of design itself. In so doing, the chapter is itself an example of one of the main points argued here, namely that the incorporation of social thinking into software practice is a wicked problem that can only be understood incrementally as solutions are tried and reflected upon.

To argue the case, this chapter is structured as follows. In the following section, I characterize why we need social thinking in software practice by defining the problem space as 'wicked'. I go on to describe our own iteration through a problem-solution space that led us to think about a shared abstraction and a metaphor of place as an appropriate response to

the difficulties we had encountered. I then outline the Locales Framework as an instance of a shared abstraction for understanding and designing application software for the support of collaboration, defining locale as the unit of analysis, and outlining the five framework aspects. Finally I reflect on our experiences in using the framework as a way of incorporating social thinking into software practice for collaborative systems. I also reflect more generally on the role of such abstractions.

Why we need 'social thinking' in software practice

In their book called "Computer Science: A Modern Introduction" (1988, first published 1982), Goldschlager and Lister characterize the computer revolution as the "augmentation of man's *mental* powers (where) the pressing of a button can cause a machine to perform intricate calculations, to make complex decisions, or to store and retrieve vast quantities of information." (*op.cit.*, p.1). The algorithm - "a sequence of steps which if faithfully performed will result in the task, or *process*, being carried out." (*op.cit.*, p.2) - is the pivotal construct that typifies traditional software practice. Requirements analysis and design techniques are largely drawn from mathematical and engineering traditions and are predicated on being able to fully specify the problem at hand and develop a specification for a system to address the problem.

In the years since, especially with the growth of the internet, this 'computer revolution' has become a 'communication revolution' and software applications are impacting more directly

on the social space of work and interaction. It is becoming increasingly accepted that traditional requirements gathering techniques are no longer adequate to account for the complex dynamic nature of this social world. While the algorithm is still important in the actual coding of systems, the emphasis in requirements gathering shifts to also understanding the social, organisational and interactional contexts in which the system will be used. This moves software practice into a different problem space - the nature of this problem space, and the ways of grappling with it, started to make more sense for me after I came across the notion of 'wicked problems'.

Wicked Problems, new challenges for software practice

The term 'wicked problem' was coined by Rittel and Weber (1973) to talk about dilemmas in social policy planning but also captured much of our own experience with software practice in CSCW. A wicked problem is usually situated in the social realm, where "the aim is not to find the truth, but to improve some characteristics of the world where people live" (*op.cit.*, p.167) - our motivation in undertaking CSCW research was to help improve the way in which people can work and interact together through the use of computer-based support. A wicked problem can never be definitively formulated and the problem is only understood progressively as solutions are developed - we had learnt a lot about the problem of supporting people working together through developing the wOrlds system. Further every instance of a general wicked problem is essentially unique - the situatedness of action. As such, there are no right or wrong solutions, only

better or worse ones. Nor are there 'one size fits all' solutions. Further, the processes for addressing the problem-solution space are inherently iterative and 'satisficing' (Simon, 1960).

This definition of wicked problems helped us to re-frame some of the general challenges for software practice. Firstly, we need new ways of thinking about software practice that take account of the intertwined problem-solution space and the iterative processes needed to learn more about the problem and come up with better solutions. Secondly, software needs to be more flexible and evolvable if we take seriously the problem-solution iterations (discussion of which is beyond the scope of this paper). Thirdly, and my focus here, we need new concepts, methods and techniques to understand and account for the social realm in which wicked problems are situated.

Many others were already trying to address variations of these challenges within their own disciplines, as were many in the CSCW community. The story that I go on with here is an instance of our own iteration through the problem-solution space of trying to bring social thinking to the design of collaborative systems to support people working together.

Towards the Locales Framework

Through work reported in the CSCW literature, our group was already aware of some of the general problems in trying to bring social thinking to software practice. In particular, some projects that brought together ethnographers and computer

scientists e.g., Bentley et al (1992), illustrated the valuable insights possible through ethnographic study of the day-to-day practices of people whose work was to be supported. These projects also highlighted the general difficulties in moving from study insights to design.

Against this background, we were also starting to learn how to incorporate social thinking into our own practice. As stated previously, we learnt of the limitations of prescriptive solutions for the support of work from experiences with workflow systems and from Suchman's (1987) discussions of the roles of plans in work practice. This led us to the wOrlds system (Fitzpatrick et al, 1995). We did not make any *a priori* assumptions about how work should happen in wOrlds, but instead provided shared spaces as settings for situated action to unfold.

We were also convinced that the social science approaches were more appropriate than traditional requirements techniques for uncovering and understanding situated work practices in complex domains. Having undertaken a course in this area, I then carried out an ethnographic-based study of a group of systems engineers with a view to understanding their needs so that we could tailor wOrlds for their use. From this we learnt two further lessons first hand.

The exact details of the wOrlds system, the study, and the lessons learnt are not directly relevant here. What is relevant is how these lessons motivated the next iteration through the problem-solution space towards the Locales Framework.

Motivating a metaphor of place

The first lesson was to do with the mismatch between the containment model of shared space inadvertently embodied in wOrlds and the ways in which the engineers worked. Many features of wOrlds, such as bounded spaces, were simulated from the physical domain, supposedly to enable people to work in 'familiar' ways. The systems engineers, on the other hand, were already exploiting features of their computing environment to work in novel ways that neither wOrlds nor their physical spaces could support.

As with many other systems based on a spatial metaphor, a wOrlds room can be characterized by notions of group related to a bounded space, where you are either in or out, and see everything or nothing. The systems engineers' work on the other hand can be better characterized by 'individuals in multiple groups' who make use of a variety of physical and virtual spaces as places of work, and where notions of relationships around centres are more relevant than containment by boundaries. (More detailed discussions can be found in Fitzpatrick et al (1996) and Fitzpatrick (1998).)

On seeing this mismatch between the instantiation of a spatial metaphor in wOrlds, and the rich ways the systems engineers used their virtual and physical spaces, I started to think about place instead of space as a guiding design principle - where place arises in the lived relationship between people and the spaces they use. Conincidentally others were also moving towards this notion of place rather than space (Harrison and Dourish, 1996).

Motivating a shared abstraction approach

The second lesson, similar to the experiences of many others in the CSCW community (Plowman et al, 1995), was that gathering, interpreting, and translating qualitative data for the purposes of systems design was difficult. Even though I came from a computing background and undertook the study with design in mind, it was still difficult to describe the work domain in a way that made sense to other technologists and that pointed to the appropriate forms of support that would work for the engineers' situation.

Reflecting more generally, CSCW is a multi-disciplinary community; those concerned with understanding and those concerned with designing tend to come from different backgrounds, neither of which fully prepares their proponents to work at the conjunction of the two areas.

On the one hand, as systems designers, we are not traditionally trained to undertake qualitative studies or to make sense of sociological accounts of work. While we might want to support the 'social' insights that emerge from others' workplace studies, it can be difficult to translate these insights into the substance of design², e.g., see Kaplan et al (1997).

On the other hand, social science traditions used within CSCW, be they ethnomethodology, distributed cognition, activity theory, and so on, were never intended to address the demands of systems design. Plowman et al (1995), for example, talk about the lack of a translation process whereby accounts of the social organization of work can be translated into design information.

A fundamental source of many of the tensions between understanding and designing could be the role of abstractions or models in the different world-views. Button and Dourish (1996) suggest that computer scientists view abstractions as *generative* in that they give rise to, as well as characterize, system action whereas ethnomethodologists (and perhaps other social scientists) view abstractions as *analytic* explanations of social action.

Solutions had thus far tended to be driven more from the social sciences - for hybrid forms of social science (Shapiro, 1994) and for more pragmatic ethnographic processes (Hughes et al, 1994).

Not wanting to engage in a discussion of the generative/analytical duality of abstractions, but having experienced the issues first hand, I started to think about whether an abstraction shared by both 'understanders' and designers could be a possible solution. The challenge was to create an abstraction that accounted for and was grounded in the sociality of work and allowed for the uniqueness of each workplace setting (the wickedness of the problem domain), while also accounting for the pragmatic needs of design and systems building.

The Locales Framework came to be this shared abstraction based on a metaphor of place.

Locales Framework

The Locales Framework, then, grew out of our own experiences and the experiences of the broader CSCW community with understanding and designing. Underpinning the evolution of the Framework was

the work of a prominent American sociologist, Anselm Strauss, and his theory of action (1993). As a non-social scientist, I found that Strauss' work gave a coherent conceptual map against which to make sense of many of ad hoc insights emerging from the workplace studies being reported in the literature³.

The central concepts of the theory that I drew upon include social worlds, action, interaction, and trajectory. A *social world* is the fundamental building block of collective action (Clarke, 1991). Members of a social world are bonded by a common, sometimes implicit, goal and perform *actions* towards the collective purpose. Actions are always embedded in *interactions*, which are continually permuting. Social worlds need '*sites and means*' to facilitate their shared interactions. An *interaction trajectory* captures the issues about how courses of action evolve over time.

Locale as the Unit of Analysis & Design

The primary unit of analysis and design in the Locales Framework is '*locale*'. Locale doesn't exist *a priori* as does a space or room. Rather, a locale is the place constituted in the ongoing *relationship* between people in a particular social world and the '*site and means*' they use to meet their interactional needs, i.e., the space together with the resources available there. As such, the framework is based on a metaphor of *place* as the lived interaction with space and resources.

In contrast to metaphors of space, which embody principles of boundaries and containment, e.g., see Roseman and Greenberg, (1996), a metaphor of place embodies principle of *centers* giving

rise to relationships (Fitzpatrick, 2000). The shared purpose of the social world, for example, provides a centre around which the people, spaces and resources make sense. With the notion of centers come other notions, for example, of relationships around the centre, of distances from the centre potentially definable along multiple different dimensions, and of dynamically varying relationships of centers to one another.

It is because locale arises in the relationship between a social world and its use of space as place, that it is potentially suitable as a shared abstraction for both understanding and designing, albeit used with slightly different emphases for each.

For understanding, the emphasis is put on people and their actions/interactions, i.e., on their appropriation and evolution of 'site and means' as place for the practical accomplishment of work. This allows for the complex, dynamic and situated 'interactional' aspects of work to be accounted for but not in isolation from where and how those interactions happen.

For designing, the emphasis is put on the 'site and means' that are used by members of the social world to meet their needs, and how better 'sites and means' or affordances could be designed for meeting interactional needs. This allows for the pragmatic needs of design, focusing on the more stable 'environmental' aspects enabling interaction but not in isolation from their use.

Locales Framework Aspects

In its current state, the Locales Framework is composed of five aspects. Each of these aspects characterizes the nature of work from a different perspective.

Locale Foundations: We first identify the group or social world of concern, looking at features such as the shared goal, how membership is defined, what are the internal structures within the group, and so on. We then identify their *locale* as constituted by the spaces, objects, tools and resources they use to support their interactions. Locale is the foundation for the rest of the framework in so far as the features identified here facilitate or constrain the life of the social world, as uncovered by the remaining aspects.

Civic Structures: This is the broader environment in which the social world and its locales exist. Depending on what is relevant, the environment could be physical, spatial, geographical, organisational, informational, professional, legislative, and so on. We can also consider the external influences on a locale, locale lifecycle processes, how locales are structured and related, and how interaction between locales is supported.

Individual Views: This accounts for the fact that groups are made up of idiosyncratic individuals who each belong to multiple social worlds simultaneously. Different members can hold different views over the one locale. Individuals also manage and negotiate different locale views simultaneously over the multiple locales in which they participate at dynamically varying levels of intensity.

Interaction Trajectory: This accounts for the social world and locale 'in action' over time: past, present and future; cycles, rhythms and phases; the performance of work and the articulation of work; routines, contingencies and breakdowns; information flows; workflows; and so on.

Mutuality: Mutuality is the very glue of collaborative activity - how *presence* is enabled in a locale and how *awareness* of that presence is supported. Mutuality enables the 'w' questions to be answered, e.g., who, what, when, where, why, and (almost a 'w') how.

These aspects are explored in detail in Fitzpatrick (1998). The motivations for choosing these aspects are described; and for each aspect, various properties are identified, together with various dimensions along which the properties could be characterized. As an example, a property of interest for a social world might be 'role'. Roles could be characterized along dimensions such as the following: fixed - fluid; formally defined - informal; imposed - socially agreed upon; assigned - self-selected; generic - specific. Questions can also be asked about how the roles arise and evolve, how they are related, how they are known about, etc.

While we draw out the five aspects explicitly as if separate orthogonal entities, they are in fact highly interdependent and overlapping. The point of the aspects is to suggest different perspectives on the same phenomenon, bringing aspects to the fore that can otherwise be forgotten. Some aspects might be more relevant or important for a particular situation or

design focus than others. Further, even if the design focus is on a particular artefact, for example, the framework can provide a background against which the artefact could be viewed in a fish-eye effect. In these ways, it is hoped we can gain a more holistic view of the domain of interest.

Using the Locales Framework

Using the Locales Framework for design generally involves a two-phase approach, as outlined in the following. This two-phase approach would then be iterative, in the fashion of all wicked problems, as the implementation of solutions sheds more light on the nature of the problems, leading to better solutions, and so on.

Phase 1 – Understanding the Current Locales The aim of the first phase is to understand the locales of interest in their current state from the perspective of the interactional needs of a social world. This could involve using the framework to structure and make sense of existing ethnographic (or other) data. Alternatively, it could involve going into the domain and directly engaging in the data gathering process.

For designers who do not have a social science background, the framework could be used as a sensitising device to help motivate initial questions and observations. For designers who do have a social science background, the framework could be used to help structure data in a form relevant to the purposes of design. For users engaged in the design process, the framework could be

used to initiate a conversation to articulate their locales of work and how these are used in their work.

For all these 'understanding' purposes, it was hoped that the framework would be sufficiently rich enough to guide the designer to look for the facilitating and constraining features in the practical accomplishment of work. At the same time, it was hoped that the framework would be general enough so as not to prescribe nor circumscribe all that is of interest so that the uniqueness of each situation could still emerge in the account of the workplace.

Phase 2 – Evolving New Locales The aim of the second phase is to explore possibilities for how existing locale(s) could be enhanced to better support the activities that take place there, or how new locales could be evolved.

This moves the focus beyond the emergent situated nature of work *per se* to the features and mechanisms of the spaces and resources of the locale that support, or have the potential to support, that work. It does this, though, in a way that is driven by interactional needs, understanding the broader context in which that interaction happens, not by *a priori* assumptions of technological solutions or by idealised abstractions of action.

The design options for meeting those needs can then be explored. The types of questions that would drive this process would be domain specific, however examples of typical questions might include the following: What interactional needs does the social world have that are poorly or inadequately supported in

their current locale(s)? How can the existing locale(s) of a social world be augmented or enhanced to better support mutuality, or individual views, or civic structures, or interaction trajectory? Can technology enable new ways of working? Can new social worlds be facilitated if computer-based spaces and resources were provided to be used as locale?

Hence, the second phase is about exploiting the strengths of any available medium, physical or computational, and the synergy among them, to better meet the social world needs identified in the first phase.⁴

In Summary

The Locales Framework was proposed as a set of concepts for describing what is as well as envisioning what could be. While it relies on qualitative methods, it has been specifically motivated by the needs of systems design. While it is moving towards systems design, design decisions are grounded in an understanding of the group's needs. Hence, the framework tries to embrace both social and technical concerns and mediate both.

It needs to be interpreted and used by designers as a tool not a recipe. It can be a sensitizing device, giving 'social thinking' pointers or heuristics for key elements of a collaborative work environment. It can also help focus on the use of space and its affordances for support of interaction, space and its affordances being the very concerns of software practice. At the same time it is sufficiently high-level, open and incomplete so as not to prescribe nor circumscribe all that is of interest. Hence, the framework is a shared abstraction in the

sense of being a common background or starting point against which domain-specific concepts and abstractions can be drawn, and design solutions can be explored.

Does the framework capture the 'right' aspects or the only possible aspects for the support of collaborative work? Surely not. As with all wicked problems, a robust dialogue between exploration of abstract frameworks and their use in practice is essential if either is to be advanced. Hence, this is an early working framework that will be evolved with use for the purposes of both analyzing work domains and designing, building and deploying systems for collaboration support within those domains. In the following section, I begin some of this reflective dialogue.

Reflections

The reflections presented here are based on my own and other colleagues' experiences in using the Locales Framework. These include the following: a study of a telehealth environment, with insights feeding into the design of new facilities for support of remote consultation (Kaplan and Fitzpatrick, 1997); a study of a group of distributed researchers (Fitzpatrick et al, 1998); a study and facilitated workshop with a government department with a focus on their information and communication flows; a heuristic evaluation of two groupware systems (Greenberg et al, 1999); the design of a generic collaborative toolkit embodying some of the principles from the Locales Framework (Mansfield et al, 1997); and various workshops describing the framework to people from industry.

On the framework..

Our experiences do suggest that the framework concepts have been able to provide a starting point for thinking about the broader contexts in which our applications will be embedded. In hindsight, this has proved most useful for software engineers who do not have a social science background and who do not want to, or know how to, engage with the intricacies of social theories - the framework can offer more than traditional software requirements techniques alone for bringing social thinking into software practice.

The framework is not an algorithmic solution, for CSCW concerns at least. As expected, this has proved to be both a strength and a weakness.

Its strength is that, in presenting such a broad brush of concepts, it can provide enough structure to support the beginnings of understanding while still allowing for the uniqueness of each workplace to demand appropriate attention. The workaday world is rich and complex, and by definition, such concepts will always be incomplete and inadequate. Remembering that every instance of a wicked problem is unique, every situation will have unique issues, concerns, needs etc. Respecting this uniqueness and complexity, the aspects are not simple distinct areas to be worked through in order as a simplifying formula; they are interlinked, and their articulation is highly incremental and evolutionary, providing a multi-faceted lens to help interpret and structure the complexity from different angles, but only as deemed appropriate. Ultimately, it

is about letting the situation itself guide and inform what are the important issues and if/how the framework can be most usefully applied. As discussed by Beck (this volume), what doesn't fit can be a valuable analytic resource.

Having said this, the framework, especially as laid out with characterizing properties and dimensions for each aspect, can usefully serve as a set of heuristics or a sensitizing checklist. A danger however is that some people might think these can be applied in a mechanistic cookbook fashion as if sufficient for 'social thinking'. Obviously this would trivialise the domain under consideration and ignore the practical accomplishment of both the work we seek to support and the work required for understanding and designing.

Its weakness, therefore, is that, not being an algorithmic solution, there is considerable effort required to creatively and intelligently interpret, evolve, extend, or even discard (aspects of) the Locales Framework for the situation at hand; hence people are integral to the process. For the more 'traditional' computer scientists I have worked with, this has proved to be problematic, explained perhaps by the more structured mathematical/engineering bias in their education. It could just be the case that some computer scientists will never want to use any such framework themselves, preferring instead that others engage in the understanding work.

It could also be the case that the processes for actually using the framework have not been explained clearly enough. Articulation of the framework to date has concentrated on laying

out the various aspects and associated concepts, not on how to apply them. To facilitate wider use, the processes for working with the framework should be made more visible and accessible. Hopefully this effort will become easier as we build up a corpus of studies and experiences.

A further weakness of the framework can be seen with regards to design. While it has been very useful for motivating design decisions at a higher conceptual level, it hasn't been as useful for informing the substance of design for specific solutions; the designer is still left to translate the motivations into actual systems. While this is better than ad hoc "implications for design" included at the end of many workplace studies, there is still a long way to go.

A design strength on the other hand is that use of the framework did prove useful for promoting design decisions that took account of the broader context of work. With locale as the unit of presentation in the design motivation, together with the different framework perspectives, we are able to tell a more complete story of what could be. For example, the design suggestions for the group of distributed researchers not only touched upon new technologies to augment the existing locale, but also new ways of using their existing physical spaces and furnishings and technologies, as well as pointing to critical inter-organisational issues.

In using the framework, the questions of how to design better locales, or what even constitutes a better locale, becomes highlighted as a significant issue. At the moment, this is

largely left to the intuition and skill of the designer. In one sense, this is not a problem since design is a creative process. The difficulty is related more to the relative newness of CSCW design as a field, and more generally the newness of software practice around interactive systems. Other creative design disciplines such as architecture have rich well-established traditions to draw upon⁵. Such design knowledge about socially-situated systems is only beginning to evolve. Are there principled approaches that could suggest what constitutes a better locale design?

On the relationship to other social thinking approaches ...

The Locales Framework is obviously not the only approach for dealing with wicked problems and for bringing social thinking into software practice. The Scandinavian/Participatory Design community has a long tradition of engaging with users in a "contextual approach to design of computer artifacts" Kyng (1998). The MUST method developed by Bodker et al (this volume) and Soft Systems Methodology (Checkland, 1981) are other similarly motivated approaches.

Other conceptual frameworks have also been developed. Hughes et al (1997), for example, proposed a number of viewpoints for presenting ethnographic data to designers. Coming more from an HCI/user-centered design background, Beyer and Holtzblatt (1998) have developed various work models (1998) for providing a structured representation of user work practices. Interestingly, there is considerable overlap in the representational concepts of

both these approaches and the Locales Framework aspects despite being developed relatively simultaneously and independently.

Perhaps the closest approach to the Locales Framework is Activity Theory. At some levels, Activity Theory can serve many of the same functions as the Locales Framework: a unit of analysis that encompasses both the social and the technical; a focus on the broader context not just on technology or artefacts in isolation; enabling description of what is as well as envisioning what could be; and so on.

On the other hand, the complexity and plurality of expressions of Activity Theory can make it difficult to come to grips with - at least that has been my experience. For the purposes of design, it is very much a theory under evolution. The efforts reflected in the work of people such as Bodker (1991), Engestrom (1991), Kaptelinin (this volume), and Kuutti (1992) are important contributions to helping make Activity Theory more accessible to software practitioners.

Perhaps the key difference between the Locales Framework approach and these other approaches is its purposeful focus on providing an open evolvable framework of concepts that can be relevant for both understanding the social situation and for relating software development to it. In this way, the Locales Framework is not just a tool for potentially making social thinking accessible for software practitioners but it also attempts to "mediate joint practice"⁶ through the development of shared abstractions.

On the primacy of abstractions or people ...

On a final reflective note, the previous discussions about the open evolvable nature of the Locales Framework, and about the critical role of people in interpreting and using the framework, beg the question of what role the shared abstractions really do play in mediating joint practice.

In attempting to account for the uniqueness of each situation and not wanting to provide a solution that prescribes or circumscribes all that is of interest, the concepts have been deliberately characterized in very open ways. But do the concepts then end up so high-level and open as to be of limited value for both understanding and designing?

To what extent does the assumption that a toolkit such as the Locales Framework could be useful for software practitioners ignore the invisible work in the skilled practical accomplishment of social thinking (Forsythe, 1999)?

How does the framework become a shared abstraction in the first place? Is it an idealized expectation? Sharing does not happen by decree but by active engagement and negotiation, but by whom and how? What is actually shared? These processes have not been explicitly addressed to date.

To what extent does the articulation of a framework such as the Locales Framework, and the assumption about it being shared, inadvertently give primacy to the conceptual tool and not the people in the process? Are the successful outcomes due more to the people involved and their sensitivities and creativity rather

than to the framework itself? Is this also the case for the other complementary approaches just noted?

Instead of looking for shared abstractions, is it 'shared people' we need, people who can engage with both the understanding of social needs and the practice of software design, who can be edge dwellers of both communities, and who can facilitate and mediate the right sorts of dialogue? How would we define or legitimize the role of such people in software practice?

These are open questions for me. I started this chapter by saying that it would be a reflection of the next phase of our own journey in trying to incorporate social thinking into our software practices to create collaboration systems that would actually be useful. I've gone on to describe the Locales Framework as if it were the product of this phase. Maybe instead what I have been describing is part of my own process towards becoming a 'shared person', an edge dweller - itself an ongoing process of 'doing' social thinking in software practice. Being a wicked problem, we'll only learn more about these questions as we try different solutions.

Acknowledgements

Eevi Beck, Victor Kaptelinin, Chris Westrup and Yvonne Dittrich have given invaluable feedback on earlier drafts of this chapter - thank you. Simon Kaplan and Tim Mansfield have been closely involved in the evolution of the work presented here. I am grateful for their insights, challenging discussions and contributions. I also acknowledge the contributions of the rest

of the wOrlds research group at UIUC and the DSTC. The work reported in this paper has been funded in part by the Co-operative Research Centre Program through the Department of Industry, Science & Resources of the Commonwealth Government of Australia.

Notes

1. The wOrlds group, led by Simon Kaplan, was located first at the University of Illinois at Urbana Champaign in the US, and was re-established at the Distributed Systems Technology Centre in Australia in 1995. In Illinois, the group members included Bill Tolone, Doug Bogia, Ted Phelps, Mark Fitzpatrick, and Annette Feng. In Australia, members included Ted Phelps, Mark Fitzpatrick, Tim Mansfield.
2. There are now some innovative programs being developed, e.g., at University of Karlskrona/Ronneby and the Information Environments program at the University of Queensland, that are attempting to educate people with a mix of social and technical skills.
3. While this theory played a critical role in the evolution of the framework, it is not needed in the application of the framework - ultimately the framework perspectives can be motivated, used and populated independent of Strauss' theory.
4. Note the focus here is not on how to re-create physical domains in the virtual realm, nor on how to make virtual interactions 'just like' face-to-face interactions.

5. For example, most 'good' homes are designed with basic components of kitchen, bathroom, living room, and bedroom. The creativity is in the detail of the components, how they are related together and how the facade of the home is presented.

6. I am grateful to Yvonne Dittrich for highlighting this point for me (personal communication, Nov 2000).

References

Beck, E. 2000. What doesn't fit: 'the residual category' as analytic resource. *This volume*.

Bentley, R., Hughes, J., Randall, D., and Rodden, T. 1992. Ethnographically-informed systems design for air traffic control. In *Proceedings of Conference on Computer Supported Cooperative Work*. (Toronto, Canada), 123-129.

Beyer, H. and Holtzblatt, K. 1998. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann, San Francisco.

Bodker, K., Kensing, F. and Simonsen, J. 2000. Changing work practices in design. *This volume*.

Bodker, S. 1991. *Through the Interface - A Human Activity Approach to User Interface Design*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Button, G., and Dourish, P. 1996. Technomethodology: Paradoxes and Possibilities. In *Proceedings of Conference on Computer Human Interaction*. (Vancouver, BC Canada), 19-26.

Checkland, P. 1981. *Systems Thinking, Systems Practice*. J. Wiley and Sons, Chichester.

Clarke, A. 1991. Social Worlds/Arenas Theory as Organizational Theory. In Maines, ed., *Social Organization and Social Process*:

Essays in Honor of Anselm Strauss. Adeline de Gruyter, New York, 119-58.

Engestrom. Y. 1991. Developmental work research: reconstructing expertise through expansive learning. In Nurminen and Weir, eds., *Human Jobs and Computer Interfaces*. Elsevier Science Publishers, Amsterdam.

Fitzpatrick, G. 1998. *The Locales Framework: Understanding and Designing for Cooperative Work*, PhD Thesis, The University of Queensland, Brisbane.

Fitzpatrick, G. 2000. Centres, peripheries and electronic communication: changing work practice boundaries. To appear in *Scandinavian Journal of Information Systems*.

Fitzpatrick, G., Kaplan, S., and Parsowith, S. 1998. Experience in Building a Cooperative Distributed Organisation: Lessons for Cooperative Buildings. In *Cooperative Buildings: Integrating Information, Organization and Architecture - LNCS 1370*. Springer, Germany, 66-79.

Fitzpatrick, G., Kaplan, S., and Tolone, B. 1995. Work, Locales and Distributed Social Worlds. In *Proceedings of European Conference on Computer-Supported Cooperative Work*. (Stockholm, Sweden), 1-16.

Fitzpatrick, G., Kaplan, S. M., and Mansfield, T. 1996. Physical spaces, virtual places and social worlds: A study of work in the virtual. In *Proceedings of Conference on Computer Supported Cooperative Work*. (Boston, MA), 334-343.

- Forsythe, D. E. 1999. "It's Just a Matter of Common Sense": Ethnography as Invisible Work. *Computer Supported Cooperative Work* 8: 127-145.
- Goldschlager, L., and Lister, A. 1988. *Computer Science: A Modern Introduction*, Prentice Hall International, London.
- Greenberg, S., Fitzpatrick, G., Gutwin, C., and Kaplan, S. 1999. Adapting the Locales Framework for Heuristic Evaluation of Groupware. In *Proceedings of OzCHI'99*. (Wagga Wagga, Australia), 30-36.
- Harrison, S. and Dourish, P. 1996. Re-Place-ing Space: The Roles of Place and Space in Collaborative Systems. In *Proceedings of Conference on Computer Supported Cooperative Work*. (Boston, MA), 67-76.
- Hughes, J., O'Brien, J., Rodden, T., Rouncefield, M., and Blythin, S. 1997. Designing with Ethnography: A Presentation Framework for Design. In *Proceedings of Designing Interactive Systems*. (Amsterdam, The Netherlands), 147-158.
- Hughes, J., King, V., Rodden, T., and Andersen, H. 1994. Moving our from the control room: Ethnography in Systems Design. In *Proceedings of Conference on Computer Supported Cooperative Work*. (Chapel Hill, NC). 429-439.
- Kaplan, S., and Fitzpatrick, G. 1997. Designing Support for Remote Intensive-Care Telehealth using the Locales Framework. In *Proceedings of Designing Interactive Systems*. (Amsterdam, The Netherlands), 173-184.

- Kaplan, S., Fitzpatrick, G., Mansfield, T., and Tolone, B. 1997. MUDDling Through. In *Proceedings of Thirtieth Hawaii International Conference on Systems Sciences*. (Hawaii), 359-548.
- Kaptelinin, V. 2000. Beyond tasks: Actions, activities, and social aspects of computer use. *This volume*.
- Kuutti, K. 1992. Identifying Potential CSCW Applications by Means of Activity Theory Concepts: A Case Example. In *Proceedings of Conference on Computer Supported Cooperative Work*. (Toronto, Canada), 233-240.
- Kyng, M. 1998. Users and computers: A contextual approach to design of computer artifacts. *Scandinavian Journal of Information Systems* 10: 7-44.
- Mansfield, T., Kaplan, S., Fitzpatrick, G., Phelps, T., Fitzpatrick, M., and Taylor, R. 1997. Evolving Orbit: A Progress Report on Building Locales. In *Proceedings of Conference on Supporting Group Work*, (Phoenix, Arizona), 241-250.
- Plowman, L., Rogers, Y., and Ramage, M. 1995. What are Workplace Studies For? In *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work*. (Stockholm, Sweden), 309-324.
- Rittel, H., and Webber, M. 1973. Dilemmas in a General Theory of Planning. *Policy Studies* 4(1): 155-169.
- Roseman, M., and Greenberg, S. 1996. TeamRooms: Network Places for Collaboration. In *Proceedings Conference on Computer Supported Cooperative Work* (Boston, MA), 325-333.

Shapiro, D. 1994. The Limits of Ethnography: Combining Social Sciences for CSCW. In *Proceedings Conference on Computer Supported Cooperative Work*. (Chapel Hill, NC), 417-439.

Simon, H. 1960. *The New Science of Management Decision*. Harper and Row, New York.

Strauss, A. 1993. *Continual Permutations of Action*. Aldine de Gruyter, New York.

Suchman, L. A. 1987. *Plans and Situated Actions: The Problem of human machine communication*. Cambridge University Press, New York.

Wastell, D. G., White, P., and Kawalek, P. 1994. A methodology for business process redesign: experience and issues. *Journal of Strategic Information Systems*, 3(1): 23-40.