

What We Talk About When We Talk About Context

Paul Dourish

Dept. Information and Computer Science
University of California Irvine
Irvine, CA 92697-3425 USA
jpd@ics.uci.edu
<http://www.ics.uci.edu/~jpd>

Abstract. The emergence of ubiquitous computing as a new design paradigm poses significant challenges for HCI and interaction design. Traditionally, human-computer interaction has taken place within a constrained and well-understood domain of experience – single users sitting at desks and interacting with conventionally-designed computers employing screens, keyboards and mice for interaction. New opportunities have engendered considerable interest in “context-aware computing” – computational systems that can sense and respond to aspects of the settings in which they are used. However, considerable confusion surrounds the notion of “context” – what it means, what it includes, and what role it plays in interactive systems. This paper suggests that the representational stance implied by conventional interpretations of “context” misinterprets the role of context in everyday human activity, and proposes an alternative model that suggests different directions for design.

1 Introduction

One of the major research directions for HCI over the past few years has exploring been the novel forms of interaction that can be achieved by integrating computer technology with the world in which we live and work. This line of research goes by a number of names – ubiquitous computing (Weiser, 1991), context-aware computing (Moran and Dourish, 2001), pervasive computing (Ark and Selker, 1999), embodied interaction (Dourish, 2001), and more. Although the nomenclature varies, the central ideas are largely the same. Extrapolating from current trends in the development of low-cost and low-power devices, ubiquitous computing proposes a digital future in which computation is embedded into the fabric of the world around us. In this world, our primary experience of computation is not with a traditional desktop computer, but rather with a range of computationally-enhanced devices – pieces of paper, pens, walls, books, hammers, etc. The opportunity implied by this ubiquitous computing vision is to capitalize our familiarity, skill, and experience in dealing with the everyday world around us. The world can become an interface to computation, and computation can become an adjunct to everyday interaction.

There are many significant research issues that this vision encompasses, but two have come to particular prominence in HCI. The first is mutual relationship between physical form and activity; how we can design computationally-enhanced devices and how their form as much as their interactive ability affects likely patterns of action and interaction. Researchers have looked towards other design disciplines to better understand the relationship between form and function and to explore the permeable boundary between physical and digital systems (e.g. Strong and Gaver, 1996; Brave and Dahley, 1997). The second concern, which will be the focus of the discussion here, is how computation can be made sensitive and responsive to the setting in which it is harnessed and used. How can sensor technologies allow computational systems to be sensitive to the settings in which they are used, so that, as we move from one physical or social setting to another, our computational devices can be attuned to these variations?

Whether we refer to it as context-aware computing or by one of its other names, the notion of “context” plays a central role in this second area of investigation. Context has been a concern in many areas of design and computer science, but it is of central importance here. One reason is straightforward: when computation is moved “off the desktop,” then we suddenly need to understand where it has gone. So, a primary concern in ubiquitous computing research is to understand the potential relationship between computation and the context in which it is embedded.

Uses of context vary. For instance, some systems encode context along with information so that it can later be used as a retrieval cue. Examples of systems that use context in this way include Lifestreams (Freeman and Gelernter, 1996), Placeless Documents (Dourish et al., 2000), Where Were We (Minneman et al., 1995), and Forget-me-not (Lamming and Flynn, 1994). A second approach is to use context dynamically to tailor the behaviour of the system or its response to patterns of use. Examples of this sort of approach include Easy Living (Brumit et al., 2000) and the Electronic Tourist Guide (Cheverst et al., 2000). Some systems, such as the Conference Assistant (Dey et al, 2001) combine both.

So, the way in which the notion of “context” has entered the realm of computational design has largely been in the form of a set of design challenges – sensor fusion, information management, system parameterization, and so forth. At the same time, designers have hoped that by incorporating notions of context into interactive technologies, those technologies can be made more sensitive to the details of specific settings of use, which social scientists have often pointed out in critiques of conventional system design (Suchman, 1987). However, the social and technical ideas sit uneasily together. In this paper, though, I want to discuss this relationship. In particular, I want to step back to consider a broader question, one which a number of researchers have recently attempted to address – just what is context, and what role is it playing in these systems? What is context doing for us, conceptually? To paraphrase Raymond Carver, What do we talk about when we talk about context?

2 Two Views of Context

My goal in this paper is to present and explore a view of context which differs in some significant ways from the conventional approach. To do this, it’s necessary first to ex-

plore how the idea of context is currently used in ubiquitous or context-aware computing practice.

2.1 Context as a Representational Problem

Software systems are representational, so a concern with context naturally leads to a concern with how context can be encoded and represented. For instance, Schilit and Theimer (1994) define context as “location and the identity of nearby people and objects.” Ryan et al (1997) define context as “location, identity, environment, and time. In one of the more extensive investigations of context-based computing, Dey et al. (2001) define context as “any information that can be used to characterize the situation of entities” and elaborate it as “typically the location, identity, and state of people, groups, and computational and physical objects.” One of the broadest definitions is one of the earliest; Schilit et al. (1994) observe, “Context encompasses more than just the user’s location, because other things of interest are also mobile and changing. Context includes lighting, noise level, network connectivity, communication costs, communication bandwidth, and even the social situation; e.g., whether you are with your manager or with a co-worker.”

Again, my concern here is not to compare and critique these specific definitions, but rather to explore the conceptual work that context is doing here. In particular, I am concerned with the assumptions and commitments that are reflected in these definitions. In particular, four assumptions seem to underlie the notion of “context” as it operates in these systems.

- First, *context is information*. It is something that can be known (and hence encoded and represented much as other information is encoded and represented in software systems)
- Second, *context is delimited*. We can, for some set of applications or application requirements, define what counts as the context of activities that the application supports, and do so in advance.
- Third, *context is stable*. Although the precise elements of a context representation might vary from application to application, they do not vary from instance to instance of an activity or an event. The determination of the relevance of any potential contextual element can be made once and for all.
- Fourth, and most importantly, *context and activity are separable*. Activity happens “within” a context. The context describes features of the environment within which the activity takes place, but which are separate from the activity itself. So, I might be engaged in a conversation, which may be happening in a location; the conversation is my activity, while the location is an aspect of the context. Broadly, although the ubiquitous computing program wants to observe that there may be fruitful relationships to be defined between context and content, they can be defined and described separately; the content or activity is “within” while the context is “without.”

These four assumptions underwrite the notion of context as it appears in much current ubiquitous computing research. In particular, the idea that context consists of a set of features of the environment surrounding generic activities, and that these features can

be encoded and made available to a software system alongside an encoding of the activity itself, is a common assumption in many systems.

In this paper, I want to discuss an alternative view of context and some of its implications. This alternative view takes a different stance of each of those four assumptions:

- First, rather than considering context to be information, it instead argues that *contextuality is a property* of information, or of objects. It's a relational property. It is not simply the case that something is or is not context; rather, it may or may not be contextually relevant to some particular activity.
- Second, rather than considering that context can be delineated and defined in advance, the alternative view argues that the scope of contextual features is *defined dynamically*.
- Third, rather than considering that context is stable, it instead argues that context is particular to each occasion of activity or action. Context is an *occasioned property*, relevant to particular settings, particular instances of action, and particular parties to that action.
- Fourth, rather than taking context and content to be two separable entities, it instead argues that context *arises from* the activity. Context isn't just "there," but is actively produced, maintained and enacted in the course of the activity at hand.

2.2 Context as an Interactional Problem

In other words, what I want to do here is to reconsider context, not as a *representational* problem but as an *interactional* problem. As a representational problem, the central concern with context is with the questions, "what is context and how can it be encoded?" What I want to argue here, by turning to social science investigations of everyday activity, is that this representational stance reflects a misunderstanding of the nature and role of contextuality in actual everyday affairs. Instead, I want to propose an interactional model of context, in which the central concern with context is with the questions, "how and why, in the course of their interactions, do people achieve and maintain a mutual understanding of the context for their actions?"

So, in this model, context isn't something that *describes a setting*; it's something that people *do*.

As a précis of my argument, one way to think about this by analogy with the notion of "relevance" in conversation. We can think of what people generally mean by context as denoting a kind of middling relevance to the matter at hand. For two people holding a conversation about a book they lately read, certain topics – such as their immediate subject of conversation, say the author of the book – are very relevant, while certain others – such as the scorer of the winning goal in the 1953 British F.A. Cup Final – are irrelevant (unless the book is about sporting trivia, of course). We would not normally describe either of these as the "context" for the conversation. Instead, the matters that we might designate as being contextually relevant are ones that may have a general bearing on the conversation and assist in its flow and construction – such as these conversants' shared history of interaction, the anticipation of an upcoming visit of the author to a local bookstore, or their appreciation of the author's other work. These are issues of intermediate relevance – neither directly relevant nor directly irrelevant. So,

similarly, the typical conception of context in technical systems is of information of a middling relevance – where an action takes place, the people at hand, and so forth.

The essential feature of this issue of relevance, though, is that the relevance of a certain matter to the conversation at hand will *emerge from the conversation as it proceeds*. The participants may change the subject or otherwise turn a matter from one of middling relevance to central relevance, or vice versa. They may suddenly realize an assumption that doesn't hold and need to backtrack to repair the conversation. They might turn the location of the conversation from “context” to “content” by remarking that it's too public and place and perhaps they should move elsewhere; and so forth. In other-wise, the determination of relevance – or of contextuality – is not one that can be made a priori. It is an emergent feature of the interaction, determined in the moment and in the doing.

In other words, context and content (or activity) cannot be separated. Context cannot be a stable, external description of the setting in which activity arises. Instead, it arises from and is sustained by the activity itself.

To provide some background to this, the next section will discuss some contributions from a particular analytic approach to social action which is centrally concerned with the detailed structure of sequences of action. On the basis of this, I will propose an alternative conception of context in ubiquitous computing and some design considerations.

3 Ethnomethodology

The particular approach to the analysis of social conduct that I will draw on here is ethnomethodology. A detailed explanation of what ethnomethodology is (and why) is not appropriate here, but let me provide a sketch in very broad strokes. (For more information, see Dourish and Button, 1998; Heritage, 1984; Garfinkel, 1967.)

Arguably, the central problem of social science is the “problem of social order”: Given that social action arises from the independent action of many individuals, how is it that it can be found, by those individuals, to be orderly, rational, or meaningful? Action in the world, at all levels, from conversations to courtroom proceedings to political campaigns, are understandable and meaningful activities, recognizably orderly in the way they proceed, even though they are outcomes of the separate and individual activities of individuals who have access only to their own mental lives. How does this work?

Broadly, we can conceive of two sources for this orderliness – orderliness from without and orderliness from within.

The first approach suggests that the orderliness of any particular instance of social action derives from a set of rules, expectations, norms and conventions that have some broad reality outside the particular occasion of the activity. So, for instance, our society might embody a set of rules about how particular forms of interaction, such as between a professor and a student, should proceed, based on a set of concepts such as power, class, dominance, deference, etc. These concepts, which we internalize in a process of

socialization, lead to a set of dispositions to act in one way or another, which in turn shape or mould specific occasions of action.

By contrast, the second approach suggests that the orderliness displayed by social action arises *from within the action itself*. It is not imposed from outside, by the action of broad rules of conduct or social norms. Instead, it is an *achievement* of social actors. Parties to some activity find, within the circumstances of action, the means to render that action meaningful and orderly; and, similarly, they make such resources available to others. In the course of a conversation, for example, participants construct their utterances so as to demonstrate that they are appropriate to the action at hand – so that they are demonstrably answers to questions, elaborations of points, interjections, agreements, questions, and so forth. Similarly, these demonstrations of appropriateness furnish to others the means to see and understand what is going on as being what is meant to be going on. So, orderliness or rationality is not an abstract property of activity in general, but rather is a property of action as enacted by specific individuals in specific circumstances. The social world is not orderly because people blindly follow social rules that are imposed on them. Instead, finding the social world orderly and meaningful is a *practical* problem that people solve, endlessly and unproblematically, as they go about their business.

4 Being Ordinary

Ethnomethodology is concerned with social action in general, and its approach to examining how orderliness of action arises from within has been applied to a wide range of forms of social action, from the conduct of science (Garfinkel et al., 1981) to crossing the street (Sudnow, 1972). One subfield of ethnomethodology, Conversation Analysis, applies the ethnomethodological perspective to the analysis of naturally-occurring conversations. Conversation Analysis seeks to uncover how the organization and meaningfulness of a spoken conduct is achieved, in real time, by the conversational participants.

The founder of the field of Conversation Analysis was Harvey Sacks. Sacks analyzed fragments of naturally-occurring conversation to uncover the unnoticed mechanisms by which people would manage the conversation as it proceeded. In his paper “On Doing ‘Being Ordinary’,” Sacks (1984) explores the question of ordinary action in a way that, first, helps to make the description of ethnomethodology above a little more concrete, and, second, casts some light on an ethnomethodological view of context.

The topic of the paper is the ordinary character of everyday activity, the very fact that most things are ordinary and unremarkable. In particular, Sacks is concerned with ordinariness as a feature of conversation, addressing the question, “how is the ordinariness of activities displayed, understood, and achieved in the course of conversation?” Recalling the discussion of relevance earlier, it’s clear that ordinariness is a crucial feature of everyday conversation. Any given utterance rests on an uncountable number of assumptions and background understandings that are part of the ordinary fabric of daily life for the speaker and hearer. So, when someone tells me, “I saw an accident on my way to the bank this morning,” I hear that the accident is something out of the ordinary, something worth reporting. Crucially, however, I do *not* question that going to the bank is something that this person might do, that they generally engage in acts of going plac-

es, that accidents are things that might happen in the sorts of places one might be when going to banks, that mornings are a time that people might be about their business, that an accident is something that can be seen, etc. These are ordinary matters, and no explanation for them need be sought or provided. If it were not possible to designate these things as simply ordinary, conversation would be impossible. So the question of ordinariness is a central one for studies of conversation.

The main element of Sacks' analysis is that activities or features of action are not simply "ordinary" in their own right, but rather are designated as (and recognized as) ordinary by social actors in the course of their activities. So, for example, if we look at conversational speech, we can see that the ways in which specific topics or events are woven into the speech serves to designate them as being ordinary or unremarkable. What people report in speech is not simply states of affairs, but the ordinary nature of those states of affairs, and this is done by the way in which they are spoken about or not spoken about. Reporting on a scene witnessed in the street also reports that the witnessing of that scene is a normal, unremarkable, ordinary thing that a speaker has a right to have done. We mutually understand, and reinforce in the course of our interactions, what it is that ordinary people do in ordinary circumstances; and it is the mutual recognition of this very ordinariness that allows us to designate activities as being ordinary, unremarkable and expected.

More broadly, this applies not only to conversation but also to action. As Sacks observes, people are unlikely to say, "Hey, let's do something ordinary tonight"; but none the less, what we do when we spend an ordinary evening is engage in activities that we know or anticipate that many other people are doing as ordinary courses of action. In both action and conversation, "being ordinary" is something that people work at, by acting in ways that they understand to be the normal activities of the groups of which they are members, and making no issue out of them in the course of their interactions. The ordinariness of everyday affairs is an accomplishment of social actors; unquestioned, unnoticed, but an accomplishment nonetheless.

Ordinary-ness, in addition, is relative to particular communities and sets of activities. The ordinary and mundane features of live for research scientists are quite different from those for bank robbers and astronauts. So, while astronauts might refer to or orient to occasions of Africa passing over their heads, and do so in ways that designate those as clearly and obviously ordinary courses of events, those same references would clearly not be ordinary things for other groups of people. Ordinariness, too, is relative to particular sets of circumstances; ordinary ways of acting on airplanes and at parties are clearly different.¹

So, three significant aspects of ordinariness emerge from this analysis. First, ordinariness is something that we do; rather than simply being a stable feature of the world, it is actively managed and achieved in the course of interaction. Second, this is a mutual

1. Famously, Garfinkel (1967) describes a set of "breaching experiments" in which he would have his students engage in actions whose ordinariness lay outside the circumstances of their production – e.g. acting in their family homes as if they were in a hotel, or stubbornly refusing to accept others' most basic statements. These experiments were designed to reveal the active processes of interpretation and production of ordinary action that underpin everyday affairs.

achievement; ordinariness must be both produced and recognized by the parties to an interaction. Third, it is relative to particular communities and activities; it is a feature of forms of competent language use for groups of language users.

4.1 Ordinariness and Context

It is, no doubt, reasonably clear why Sacks' discussion of ordinariness is relevant here. Ordinariness is an occasioned property of action in just the same way as context. Like ordinariness, context is managed moment by moment, achieved by those carrying out some activity together, and relative to that activity and to the forms of action and engagement that it entails. Sacks' discussion of ordinariness as an achievement of social actors illustrates what I mean by turning from something as a "representational" problem to an "interactional" problem.

Having outlined this alternative model of context, I want to go on now to consider some of its consequences for interactive system design. To do this, I need to place things in a larger frame.

5 Practice

I have argued that context is a feature of interaction. The traditional concern in context-based systems is to be able to use the context in order to discriminate or elaborate the meaning of the user's activity. If activity is the site of contextual manipulation, then this move argues for a link between action and meaning as the primary concern of ubiquitous computing. We find the link between these two in the concept of practice.

In HCI and CSCW, we are familiar with the concern with work practice, as expressed by ethnographic and other "rich description" approaches to studies of working settings (Plowman et al., 1995; Luff et al., 2000). In these areas, the informal understanding of practice is "the detail of what people actually do" (in contrast to the formal procedures that organizational rulebooks might suggest that people *ought* to do.) The concern with designing software systems around the details of real needs and real use has led to an interest in techniques that can help uncover practice.

In his book "Communities of Practice," Etienne Wenger puts forward a richer notion of practice. For Wenger, practice is not merely about what people do, but about what they experience in the doing. "Practice," he explains, "is first and foremost a process by which we can experience the world and our engagement with it as meaningful." (Wenger, 1998:51)

Wenger's focus on meaning comes from his particular interest in learning within communities of practice. As an example, take the case of an apprentice shoemaker. On his first day on the job, he encounters a host of different pieces of leather, but beyond basic characteristics like shape and color, he has no way to discriminate between them. They simply do not mean very much to him. However, as he becomes more skilled, being gradually brought into the center of the community of practice, he slowly learns to see the leather as a shoemaker sees it. His experiences and developing skills teach him new ways to assess, understand, and encounter the leather. He begins to understand it as it relates to the practice of shoemaking. It takes on new meaning in terms of its du-

rability, workability, weaknesses, and the opportunities it holds for particular sorts of use in the hands of a skilled shoemaker. The same sort of argument applies beyond the obvious examples of physical skill. So, for example, the process of computer science education involves teaching students not just how computer systems work, but how to see the world as computer scientists do. For example, it teaches people a stance from which the world reveals itself to them as available for certain kinds of representation and automation. Practice – the practice of being a computer scientist, or the practice of being a shoemaker – is about finding the world meaningful, in terms of the actions that it affords. As one acquires these skills, new aspects of the environment become relevant for the activities that one performs. The scope of potential context broadens.

So, the concept of practice is one that unites action and meaning. It describes how the world reveals itself to us as one that is meaningful for particular sorts of actions, and as a result of our participation in communities. As competent social actors in particular domains, we can find the world and the settings we encounter as meaningful. This unification of action and meaning is also central to the question of context, since context is essentially about the ways in which actions can be rendered as meaningful – how a particular action, for example, becomes meaningful for others by dint of where it was performed, when, or with whom. The questions raised earlier – what constitutes context, how people may orient to features of the world as contextual or central, how relevance is managed, etc. – are questions of practice.

What is crucial to the interactional (rather than representational) view is to see practice as a dynamic process. It evolves and adapts. As technologists, then, our concern is not simply to support particular forms of practice, but to support the evolution of practice – the “conversation with materials” (Schön, 1983) out of which emerges new forms of action and meaning.

This is no mean challenge. Nor is it a single, distinguishable goal. Rather, it is a broad agenda for research in interactive system design. As a result, then, no absolute solutions can be presented here. Instead, I will give examples that serve to illustrate the potential design consequences of a primary concern with practice and evolution.

6 Practice and Technology

As discussed above, the representational view of context arose as a response to practical problems on the part of system designers. So, if practice is the lens through which we can resolve the problems of context, then what does that imply for the design of new technical artifacts and systems?

One starting point is to set out some general principles that govern this form of interaction. Some of these are outlined elsewhere (Dourish, 2001). Here, I will put forward just one, and illustrate it with some examples.

By turning our attention from “context” (as a set of descriptive features of settings) to “practice” (forms of engagement with those settings), we assigned a central role to the meanings that people find in the world and the meanings of their actions there in terms of the consequences and interpretations of those actions for themselves and for others. The important point, however, is that we now see those meanings as essentially

open-ended; we recognize that part of what people are doing when they adopt and adapt technologies, incorporating them into their own work, is creating and communicating new meanings through those technologies as their working practices evolve. So, the principle we need to bear in mind here is that *Users, not Designers, Control Meaning*. This means that meaning emerges at a point of use, rather than being predefined by the system developer. So, we incorporate this principle into system designs by exploring technologies in which interpretation and meaning are aspects of how the systems are used, moment-to-moment.

Here are three examples, drawn from ongoing work.

6.1 Systems that Display Their Context

This design principle draws our attention to the way that users of computer systems need to form interpretations of the state of the machine in order to determine appropriate courses of action. Hutchins et al. (1986) identifies the “gulf of interpretation” – the difficulty of interpreting the system’s state as a response to the user’s command – as one of the major problems that HCI design attempts to address, but this problem extends beyond simply the execution of commands. Especially in modern networked systems, matching action to the state of the system can be extremely problematic. Typical interfaces systematically hide the details of their operation, even though those details are frequently crucial to an evaluation of the current state of the system. Has the system paused because it’s working, because of a network problem, or because of low memory?

One solution to this problem is to consider how a system can display aspects of its own context – its activity and the resources around which that activity is organized. Looking again at the issue of practice, the goal is to allow the system to reveal a richer picture of activities, and so provide users with a more nuanced interpretation of the meaning of the system’s action (Dourish, 1995).

As an example, one domain where we are currently exploring this approach is in that of security of networked systems. Technical investigations of security have provided us with systems that can offer mathematically provable encryption, zero-knowledge proofs, etc. For users, though, the central question is, “is this system secure *enough* for the task I need to perform now?” Two things are striking about this question. First, that “secure” and “secure enough” are quite different things; and second, that most interactive systems do not provide end users with the means to make a reasonable determination of the degree of security currently available.

Current work is investigating the extent to which visualization techniques can provide users with a dynamic understanding of the security configuration of their system and its applications, and thus provide them with the resources to make these sorts of determinations. This is clearly not a question of simply defining degrees of security, nor of prescribing secure and insecure operations; but rather, allowing the user to make continual determinations of the potential consequences of their actions and their opportunities to reconfigure or realign the technologies through which they are conducting their actions.

One critical issue here is the *dynamic* nature of security and configuration information. This is especially important in mobile settings. Most laptops, for instance, are used

to connect through many different network arrangements, from the office, home, or on the road. Similarly, handheld devices move through environments in which they are handed off from one network infrastructure to another. The flipside of the “seamless access” to which most mobile systems aspire is the systematic hiding of network heterogeneity; but it is precisely that heterogeneity that determines users’ assessment of security. While maintaining seamless access, our approach aims to visualize aspects of this heterogeneity, and so allow users to explore the technical context in which their applications operate.

6.2 Architectures for Adaptation

One of the consequences of how information systems are incorporated into different settings and acquire different meanings in the course of their use is that they are adapted to the different settings in which they are used. A number of researchers have studied aspects of tailorability (how systems can be adjusted by their users to suit individual needs) and adaptation (how systems can infer patterns of actions of users and adapt themselves to them). Examples include the work of Mackay (1990), MacLean et al. (1990), and Mørch and Mehandjiev (2000). However, although they are strongly related, these two areas of work focus on different issues. Tailorable systems place control in the hands of users, but frequently offer control only over surface-level features of the system’s behaviour, such as user interface structure and appearance, short cuts, and defaults. In turn, adaptive systems may address more foundational issues (including component distribution, optimization, etc.) but generally operate autonomously. One of our interests is in what sorts of infrastructures allow for user-initiated customization of “deep” software structures.

In prior research, I have investigated the use of “reflective” architectures to support this sort of deep customization (Dourish, 1998). Current investigation is focused on less computationally expensive mechanisms for achieving similar ends, specifically for supporting “migratory” applications – ones that move from place and from device to device. Our infrastructure, called Emigre, uses run-time XML-based architectural descriptions (Dashofy et al., 2001) to allow applications to reconfigure themselves to local technical conditions (e.g. device and bandwidth availability) and patterns of use (e.g. synchronous or asynchronous interaction). The important point, again, is that the internal system structures that affect how the system can be used are available for some degree of user inspection and manipulation, and that this reflective activity be fairly seamlessly integrated with conventional use, by analogy with the way that everyday objects are available to use both as artifacts and tools simultaneously.

So, in this approach, the system’s own structure and action become available as context for making decisions about adaptation and configuration of device resources.

6.3 Structures in Information Spaces

In information work, the meaningfulness of information for people’s work is often encoded in the structures by which that information is organized. For example, Bowker and Star (1999) have provided a detailed exploration of the way that structures of classification and their use reflect and embody tacit understandings of the meaning of information for particular groups. Similarly, in some work on the use of classification

structures in engineering work, we observed the ways that information structures in a single organization were intimately tied to particular forms of use, causing problems for the mutual intelligibility of information (Dourish et al., 1999).

We are interested in the ways in which we can provide information workers with interfaces that give them a direct experience of the structures by which information is organized, especially in collaborative settings. Drawing on research into spatial hypertext (Marshall and Shipman, 1993), we are building systems for collaborative sense-making in which structures are directly accessible as part of an information-centric model of interaction. In traditional hypertext systems, such as on the Web, users organize information by creating explicit structures of links from one item to another. By contrast, spatial hypertext systems allow users to organize information in free-form two-dimensional space, much like layout out cards on a table; by drawing on principles of human perception, the system analyses the layout of the information to find structures that are visually salient to users, such as clusters, piles, columns, tables, etc.

This style of interaction has a number of valuable properties, but a couple are relevant here. The first is that users interact directly with the information objects rather than with abstract structures by which those objects are organized. There is no separation between two levels of control. The second is that structure emerges in the course of a users' interaction, rather than having to be specified all at once or in advance of the actual data being incorporated. The third, in a collaborative setting, is that users negotiate the structures by which information will be organized, incrementally building on each other's work in creating a collective organization for information that reflects their immediate needs, concerns, and understandings.

As in the previous examples, this approach to information organization sees the structure and meaning of interaction arising out of the detail of that interaction itself. The system is specifically designed to allow forms of practice to emerge and evolve, rather than requiring users to fit their work and their information into predefined patterns.

7 Context and Embodied Action

What I have been presenting here is an alternative view of context in interaction. This is not simply a different technical model or a different design for how context can be represented. Rather, it is a different proposal about what context *is*. The difference is significant. In particular, it undermines the idea that, somehow, by encoding contextual information, interactive systems can be brought closer to the situated or embedded models suggested by social analysts (Suchman, 1987). This project fails because it relies on a set of assumptions about context – the assumptions outlined in §2.1 – that do not hold in practice.

In contrast, I have presented a model of context here which in which context and activity are mutually constituent. This approach is one that I have been calling “embodied interaction.” The essential feature of embodied interaction is the idea, as illustrated above, of allowing users to negotiate and evolve systems of practice and meaning in the course of their interaction with information systems.

One feature of the examples that might seem unexpected is that they do not simply deal with the traditional domain of context-aware computing – mobile devices embedded in sensor networks, responding to different situations of use. This is certainly unusual within the frame of reference adopted by the “context-aware” applications discussed at the beginning of this exploration. However, the reason for this transition may be clearer in terms of the discussion as presented here. Embodied interaction encompasses more than simply physically-available interfaces. Embodiment is not about physical reality, but rather about availability for engagement. The embodied-interaction perspective is concerned with the way in which the meaningfulness of artifacts arises out of their use within systems of practice.

So, embodied interaction encompasses but extends beyond the physical, and so beyond the scope of physically-based ubiquitous computing or tangible interfaces (Ishii and Ullmer, 1997). The principles underlying these interfaces – in terms of how they mediate meaningful interaction between individuals – are broader ones, and the role of context in interactive systems is broader than that conceived of by many definitions of “context-aware computing.” The importance of context is not what it *is* but what it *does* in interaction – the role that it plays and the ways in which it is sustained and managed. The physical world is one in which we can see this at work, but by no means the only one. So, those who identify context-aware computing as a critical agenda for the development of interactive systems are, clearly, quite right to do so; where they may be less sure-footed is in defining context-aware computing as the responsiveness of interactive technologies to predefined features of the times and places in which it is used.

8 Conclusions

“Context” is a slippery notion. Perhaps appropriately, it is a concept that keeps to the periphery, and slips away when one attempts to define it. The goal here has not been to define it, but rather to ask what “work” the term is doing as it is used in contemporary research in HCI. In particular, I have been concerned here with approaches to context that assume some stable separation between the “context” and “content” of activities that people might perform. This separation is at the heart of proposals to capture or represent context, which is clearly a meaningless claim if we cannot nail down at least something that is to be captured or represented. However, studies of everyday action and interaction seem to provide grounds to question this assumption.

In addition, moving away from this representational view of context lets us see that the same basic insights into how context operates are applicable beyond the traditional embedded-interactive-device model of ubiquitous computing. Indeed, if we take “ubiquitous computing” seriously, then we should be applying its ideas ubiquitously, not just in the relatively narrow areas of interaction with handheld and embedded devices.

The title of this paper is adapted from that of the Raymond Carver short story, “What we talk about when we talk about love.” The genius of Carver’s work is in its detailed exploration of the mundane reality of everyday life. The approach outlined here also takes the mundane details of lived experience as the basis for understanding context, not as a stable description of the world, but as the outcome of embodied practice.

Looking at everyday action, then, pays off in two ways. First, it brings to our attention a set of problems about the ways in which context is conceived of in current design practice. Second, it provides us with a potential solution by furnishing us with the means to understand where our attention might instead be directed. The problem is not that context does not matter; it matters a great deal. Rather, the problem is that context is being continually renegotiated and defined in the course of action, and through this negotiation, the actions that individuals undertake can become intelligible and meaningful to each other.

Context is a central issue for HCI design and for interactive systems more broadly. The goal of the work described here is to find the right scope of the problem.

Acknowledgements

The ideas expressed here have benefited from discussions with many people, including Tom Moran, David Redmiles, Dave Snowdon, and Jack Whalen. I am also grateful to participants at the FX PAL Workshop on Mobile Services and the HCI Consortium meeting on Pervasive Computing for their contributions.

References

- [1] Ark, W. and Selker, T. 1999. A Look at Human Interaction with Pervasive Computers. *IBM Systems Journal*, 38(4), 504-507.
- [2] Bowker, G. and Star, L. 1999. *Sorting Things Out: Classification and its Consequences*. Cambridge, MA: MIT Press.
- [3] Brave, S. and Dahley, D. 1997. inTouch: A Medium for Haptic Interpersonal Communication. *Proc. ACM Conf. Human Factors in Computing Systems CHI'97 (Atlanta, GA)*. New York: ACM.
- [4] Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. 2000. EasyLiving: Technologies for Intelligent Environments". *Proc. Conf. Handheld and Ubiquitous Computing HUC'00*. Berlin: Springer Verlag.
- [5] Cheverst, K., Davis, N., Mitchell, K., Friday, A. and Efstreatiou, C. 2000. Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences, *Proceedings of ACM Conf. Human Factors in Computer Systems CHI 2000*, 17-24, New York, NY: ACM Press.
- [6] Dashofy, E., van der Hoek, A., and Taylor, R. 2001. A Highly-Extensible, XML-Based Architecture Description Language. *Proceedings of the the Working IEEE/IFIP Conference on Software Architecture (WICSA 2001)*, Amsterdam, The Netherlands.
- [7] Dey, A., Abowd, G., and Salber, D. 2001. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2-4).
- [8] Dourish, P. 1995. Accounting for System Behaviour: Representation, Reflection and Resourceful Action. *Proc. Third Decennial Conf. Computers in Context CIC'95 (Aarhus, Denmark)*.
- [9] Dourish, P. 1998. Using Metalevel Techniques for Flexibility in CSCW Toolkits. *ACM Trans. Computer-Human Interaction*, 5(2), 109-155.

- [10] Dourish, P. 2001. *Where the Action Is: The Foundations of Embodied Interaction*. Cambridge: MIT Press.
- [11] Dourish, P. and Button, G. 1998. On Technomethodology: Foundational Relationships between Ethnomethodology and System Design. *Human-Computer Interaction*, 13(4), 395-432.
- [12] Dourish, P., Lamping, J., and Rodden, T. 1999. Building Bridges: Customisation and Mutual Intelligibility in Shared Category Management. *Proc. ACM Conf. Supporting Group Work GROUP'99* (Phoenix, AZ), 11-20.
- [13] Dourish, P., Edwards, K., LaMarca, A., Lamping, J., Petersen, K., Salisbury, M., Terry, D., and Thornton, J. 2000. Extending Document Management Systems with User-Specific Active Properties. *ACM Trans. Information Systems*, 18(2), 140-170.
- [14] Freeman, E. and Gelernter, D. 1996. Lifestreams: A Storage Model for Personal Data. *SIGMOD Record*, 25(1).
- [15] Garfinkel, H. 1967. *Studies in Ethnomethodology*. Cambridge: Polity.
- [16] Garfinkel, H., Lynch, M., and Livingson, E. 1981. The Work of a Discovering Science Construed with materials from the Optically Discovered Pulsar. *Phil. Soc. Sci.*, 11, 131-158.
- [17] Heritage, J. 1984. *Garfinkel and Ethnomethodology*. Cambridge: Polity.
- [18] Hutchins, E. L., Hollan, J. D., and Norman, D. A. 1986. Direct manipulation interfaces. In Norman & Draper (Eds.), *User Centered System Design: New perspectives on human-computer interaction*. Hillsdale, NJ: Erlbaum.
- [19] Ishii, H. and Ullmer, B. 1997. Tangible bits: Towards seamless interfaces between people, bits, and atoms. *Proc. ACM Conf. Human Factors in Computing Systems CHI'97* (Atlanta, GA), 234-241. New York: ACM.
- [20] Lamming, M. and Flynn M. 1994. Forget-me-not: Intimate Computing in Support of Human Memory. *Proceedings of FRIEND21 '94, International Symposium on Next Generation Human Interface*.
- [21] Luff, P., Hindmarsh, J., and Heath, C. (eds) *Workplace Studies: Recovering Work Practice and Informing System Design*. Cambridge: Cambridge University Press.
- [22] Mackay, W. 1990. Patterns of Sharing Customizable Software. *Proc. ACM Conf. Computer-Supported Cooperative Work CSCW'90* (Los Angeles, CA). New York: ACM.
- [23] MacLean, A., Carter, K., Moran, T., and Lövstrand, L. 1990. User-Tailorable Systems: Pressing the Issues with Buttons. *Proc. ACM Conf. Human Factors in Computing Systems CHI'90* (Seattle, WA). New York: ACM.
- [24] Marshall, C. and Shipman, F. 1993. Searching for the Missing Link: Discovering Implicit Structure in Spatial Hypertext. *Proc. ACM Hypertext '93 Conference*, 217-230. New York: ACM.
- [25] Minneman, S., Harrison, S., Janssen, B., Kurtenbach, G., Moran, T., Smith, I. & van Melle, W. 1995. A Confederation of Tools for Capturing and Accessing Collaborative Activity. *Proc. ACM Conf. Multimedia'95* (San Francisco, CA). New York: ACM.
- [26] Moran, T. and Dourish, P. 2001. Context-Aware Computing: A Special Issue of *Human Computer Interaction*, 16(2-4).
- [27] Mørch, A. and Mehandjiev, N. 2000. Tailoring as Collaboration: The Mediating Role of Multiple Representations and Application Units. *Computer Supported Cooperative Work* 9(1), 75-100.

- [28] Rodden, T., Cheverst, K., Davies, K., & Dix, A. (1998). Exploiting context in HCI design for mobile systems. Proceedings of the Workshop on Human Computer Interaction with Mobile Devices. Glasgow, Scotland.
- [29] Plowman, L., Rogers, Y. and Ramage, M. 1995. What are Workplace Studies For? Proc. European Conf. Computer-Supported Cooperative Work ECSCW'95 (Stockholm, Sweden), 309-315. Dordrecht: Kluwer.
- [30] Ryan, N., Pascoe, J., and Morse, D. 1997. Enhanced Reality Fieldwork: The context-aware archeological assistant. In Gaffney, Leusen, and Exxon (eds), Computer Applications in Archeology.
- [31] Sacks, H. 1984. On doing 'being ordinary'. In Atkinson and Heritage (eds), Structures of Social Action, 413-429. New York: Cambridge University Press.
- [32] Schilit, B., Adams, N., and Want, R. 1994. Context-Aware Computing Applications. Proc. IEEE Workshop on Mobile Computing Systems and Applications (Santa Cruz, CA). New York: IEEE.
- [33] Schilit, B. and Theimer, M. 1994. Disseminating active map information to mobile hosts. IEEE Network, 8(5), 22-32.
- [34] Schön, D. 1983. The Reflective Practitioner: How Professionals Think in Action. New York: Basic Books.
- [35] Strong, R. and Gaver, W. (1996). Scent and Shaker: Supporting Simple Intimacy. Proc. ACM Conf. Computer-Supported Cooperative Work (Short Papers), 29-30.
- [36] Suchman, L. 1987. Plans and Situated Actions: The Problem of Human-Machine Communication. Cambridge: Cambridge University Press.
- [37] Sudnow, D. 1972. Temporal Parameters of Interpersonal Observation. In Sudnow (ed), Studies in Social Interaction, 229-258. New York: Free Press.
- [38] Weiser, M. 1989. The Computer in the 21st Century. Scientific American.
- [39] Wenger, E. 1998. Communities of Practice: Learning, Meaning, and Identity. Cambridge: Cambridge University Press.