

Where the Action Is: The Foundations of Embodied Interaction

Paul Dourish

MIT Press, December 2001, ISBN 0-262-04196-0.

Most textbooks in HCI and CSCW do not offer a coherent and over-arching understanding of social and technological issues. They present a variety of techniques and technologies, and outline a little history, but offer little in terms of theory that addresses the complexity of collaborative systems' structure and use. The majority of practitioners and researchers do not see theory as one of the things that they do. Immersed in their craft, focusing on technological innovation or ethnomethodological detail, they do not engage in theoretical abstraction. Technologists contentedly explore new tools and devices, with little heed to older disciplines or deeper discussion about the limits and assumptions inherent in their craft. The area of sociology most influential in CSCW, ethnomethodology, deliberately keeps theorising and generalisation at a distance, seeing abstraction as brutish and creativity as foreign. In our field, theory is like the public library. If asked, most of us would say that we are glad that it is around—but few of us actually go there. Most see it as a haven for the old, the unemployed and the eccentric. Paul Dourish is a card-carrying member, however. He reads avidly and widely, but is also a skilful system designer and developer. The result is a book that is deep, accessible and useful, which is a rare thing nowadays.

In the introduction, Dourish lays out the structure of the book. He describes and reflects on two current trends in system design: tangible and social computing. By looking at the usually hidden assumptions in system design, he makes his later theoretical discussion more relevant and accessible to a computer science audience. He does not develop new philosophy or social theory, but draws upon established 20th century philosophy of language and phenomenology. Much of this material is likely to be unfamiliar to 'the average programmer' and, although Dourish presents it well, it may still be challenging to the reader. However, its use is one of the book's main contributions. He uses it to ground a conceptual framework and a corresponding set of principles for system design practice. He aims to do justice to the sociality and heterogeneity of interactive media, and to avoid putting theory above practice—or vice versa. His ideal is balance: "the ability to develop systems that resonate with, rather than restrict (or, worse, refute), the social organization of action".

The term 'tangible computing' is used very broadly in this book. The chapter on this theme covers the designs of Hiroshi Ishii's group at MIT's Media Lab, but also ubiquitous computing work such as the badges, tabs and LiveBoards of Xerox PARC, and augmented reality systems such as Pierre Wellner's DigitalDesk. Although 'UbiComp' is more fashionable nowadays, Dourish chooses a term that helps him focus more on our perception and "the ways we experience the everyday world" than on computation and technology. Much of this discussion centres on exemplary systems, and the way that an increasing number and variety of computational devices and sensors are distributed in our environment. This contrasts with older systems that used very few media and which were encapsulated in the beige box of the traditional PC.

The following chapter is on social computing: "the application of sociological understanding to the design of interactive systems". This stands in contrast to the more traditional tendency for designers to treat people as isolated system users, with little account taken of organisational and social context. Dourish draws upon some influential studies of existing technology such as that of an air traffic control room by Hughes et al., and the ethnography of a print shop by Bowers, Button and Sharrock. Compared to the previous chapter, exemplary systems are scarce. There are none of the previous chapter's attractive images of exotic displays and devices, so beloved in undergraduate lectures and conference presentations, as this chapter addresses systems' internal structure rather than external interaction. Dourish has developed more systems in the 'social' category than the 'tangible', and here he offers as examples Mansfield et al.'s Orbit and some of his own work: using a system design technique, computational reflection, to offer users an account of deep system structure. In describing this particular technique, Dourish touches on the essence of the entire book:

What is radical is the relationship it proposes between technical design and social understandings. It argues that the most fruitful place to forge these relationships is at a foundational level, one that attempts to take sociological insights into the heart of the process and fabric of design. (*p.* 87)

What he proposes is not only conceptual but also architectural. Raising HCI's awareness of sociological concepts and issues, and using them in requirements capture, is not enough in itself. Similarly, UbiComp and tangible computing can only take us so far. They extend the set of external media used by computers, to better match the media and senses used in human activity. However, even if there is a sensor in every paving stone and a display in each kitchen utensil, systems' behaviour will remain reductionist and unaccountable unless we change the design principles for internal representations, especially with regard to categorisation and adaptation.

Having discussed the design issues raised by tangible and social computing, Dourish sets these issues within more general discourse on representation, interpretation and design. In recounting the major conceptual shift in 20th century philosophy, Dourish carves out the touchstones of his framework and design principles. What tangible and social computing share is their move towards a better fit with everyday human activity, understanding and interaction. One of the most significant lessons that we are re-learning is the practically engaged and non-rationalising way that everyday

activity takes place, which again stands in contrast to the planned and rational abstraction assumed by much of the older work in HCI. As Dourish points out, this characterisation of everyday activity is rarely used in HCI but familiar to academic communities with wider reading habits as the 'readiness –to–hand' of Heidegger. A tool such as a hammer is ready–to–hand when, after using it for some time, one is familiar enough with its use to stop having to maintain conscious awareness of how one is using it while one uses it. Instead, one becomes engaged in the practical activity of hammering. Similarly, when speaking in a normal, everyday way, one doesn't have to rationally plan and control each word while saying it. The same is true for the other utterances, gestures and symbols–in–use that collectively constitute everyday activity. Dourish also draws on Schutz' elucidation of the social or intersubjective element of everyday perception and activity, and Merleau–Ponty's discussion of the way that the body, through the interwoven senses, plays a vital role in everyday perception. He briefly touches on Wittgenstein's philosophy of language. He makes one of the finest index entries of recent years —Wittgenstein, Ludwig; and Elvis Presley, 122—and quotes the famous line that encapsulates so much of Dourish's approach: The meaning of a word is its use in the language.

Everyday human interaction is embodied in that it is non—rationalising, intersubjective and bodily activity. Philosophers, sociologists and other writers have long discussed the way that much of our activity has these characteristics, but Dourish's point is that traditional approaches to HCI design do not take full account of embodiment, i.e. they are not in accord with the activity they aim to support. More generally, the everyday practice of HCI is as non–rationalised and uncritical as those whose activity it aims to support. Most designers 'just do it', without much thought as to the implicit theoretical assumptions underlying their approach to design, and the strengths and limitations of those assumptions. Computer science is repeating the 'learning experience' that philosophy, linguistics, sociology, architecture and many other fields went through in the first half of the last century. We need the same conceptual shift in computing, and our young field has to learn from old masters, in particular Heidegger and Wittgenstein.

Dourish focuses on raising our awareness of embodied interaction, or interpretation of a system by a user as ready–to–hand. He presents traditional HCI design as being based on its opposite, i.e. rationalising, objectifying and abstracting activity, or interpretation by the user as present–at–hand. He later discusses the shift between these two modes of interpretation as varying the degree of coupling between the interpreter and the system. As he puts it on p. 139, the existence of both modes is critical to the effective use of technologies. However, Dourish's approach swings from one extreme to the other, neglecting presence–at–hand and focusing almost entirely on embodied or ready–to–hand interaction. He doesn't address the relationship between the two modes, and how we move between them. If there is an overall weakness in the book, this is it. Dourish is justified in his work to establish that embodied interaction is underemphasised and under–supported in system design, and this work is a significant contribution, but some interaction is not embodied and it is important that it is not.

Heidegger, and his successors such as Gadamer and Ricoeur, held that situations where a system, tool or symbol becomes present–at–hand may be crucial to the individual's learning and to the differences between individuals. The ongoing 'feedback loop' of interpretation and understanding integrates these two modes, and affords variation in people's understanding as well as consistency in their behaviour. For example, creativity can be considered as the variation of an individual's subjective understanding from his or her prior understanding and from others'. The individual may then be very conscious of his or her own activity, rationalising it and very aware of it, i.e. the system, tool or symbol is present–at–hand. With experience of its use, however, it may become understood and familiar, i.e. more ready–to–hand and embodied. Similarly, as two people perceive one another's use, with each interpreting and reacting to each other, they can achieve intersubjective consistency of behaviour. A use or activity that is new and present–at–hand for one of them can thus become learned and ready–to–hand for both. The circular process of interpretation, whereby perception and activity are influenced by understanding, but also feeding onto and changing understanding, thus relies on the interplay between ready–to–hand and present–at–hand interpretation. Embodied interaction, as Dourish's book aims to make clear, is an aspect of human activity that is underemphasised in HCI. Nevertheless, ready–to–hand embodied interaction and present–at–hand objectification are interdependent—and the book does not address this.

This imbalance or incompleteness in his approach makes itself felt through his consistent opposition of 'computational' and 'digital' with 'real' and 'physical'. New technologies are usually treated as 'unreal' as they don't operate in ways that are familiar and embodied—yet. When the interaction involving them is unfamiliar, cutting across distance and people in ways not yet integrated into the everyday patterns of social activity, insufficient acceptance of the way that what is present–at–hand changes to be ready–to–hand leads to this awkward or contradictory terminology.

Dourish suggests that we can use the idea of embodied interaction in two ways. The first is in better understanding human interpretation, in particular the way that people's interaction with systems is a fundamentally embodied phenomenon (p. 145). The second use of embodied interaction is as a critical stance to take in discussing the design of existing technologies. In a way that echoes the theoretical imbalance mentioned in the previous two paragraphs, the book emphasises approaches to computing that are in accord with embodied interaction or readiness–to–hand, such as tangible and social computing. Dourish does not go so far as to address the way that systems exemplifying more reductionist and objectifying approaches to interpretation and meaning, such as web search engines and organisational workflow systems, are also part of the everyday use of computers. Their use is tightly bound in with the use of tools whose foundations are more in the style of embodied interaction. An open challenge for system design, and for CSCW in general, is to address the interdependence and integration of the many subtle shades from objectification to subjective and contextual interaction.

While writing the book, Dourish was designing and building a system that might have served him well here: Placeless. He has said that he might not have had the distance from Placeless that would have let him write about it, but this document system took an important step forward. It used fine-grained and flat models that could be dynamically and contextually re-interpreted to mimic or reconstruct multiple 'objective' modelling schemes, such as a hierarchical file systems, taxonomies of document content and organisational workflow models. Rather as Huxley suggested that God had an inordinate fondness for beetles, computing has an inordinate fondness for hierarchies. Placeless makes no one of them primary, but supports and interrelates many of them. It also treats such hierarchies in the same way as more obviously subjective models such as personal annotations on documents. It tailors its re-interpretation to suit the identity of the user, his or her individual view of documents, role in the workflow and choice of document tools, and the more objective categorisation schemes those tools depend on. To some extent, it lets people use separate tools in a way that integrates them, and supports their interdependence.

A third possible use of embodied interaction, which Dourish does not suggest, is as a means to directly create new designs for deep system structure, especially when implementable features not present in current designs are seen in humans' embodied interaction. In the later parts of the book, there is a tendency to step away from the topic of deep structure. An example of this is in his discussion of ontology and intersubjectivity. He points out that the term 'ontology' has some currency in software engineering, although not quite in a philosophical sense. One recent paper, written by researchers well-respected in their area of computing, suggests that "an ontology is a conceptualisation of a domain into a machine readable format". This is clearly far from the notion of ontology that Heidegger put forward, and which Dourish echoes, that treats ontology as a dynamic phenomenon emergent from participative practice (p. 130). However, this is only one ontological approach. 'Designing an ontology' only seems odd until one remembers the means by which a computer system represents and participates in the world, and the way that the computer succeeds or fails in establishing any kind of 'intersubjectivity' with regard to an accord between its activity and user activity. Dourish discusses the intersubjectivity between designer and user, and that between users, but not that between user and computer. As a description of human interpretation, the notion of ontology used in software engineering is clearly weak, static and positivist. Dourish makes it clear that much of HCI's traditional approach to human activity has such an inaccurate but methodologically convenient notion at its heart. Unsurprisingly, however, this notion is quite accurate as a description of the weak, static and positivist representations and interpretive processes of current software designs. Another famous phrase of Wittgenstein comes to mind: "The limits of my language are the limits of my world." This is true for the computer, too, and the origin of what Dourish and Button called the 'paradox of technomethodology' is the difference between these two approaches to ontology. Until we develop new, open and adaptive forms of deep structure that are methodologically convenient for designers as well as dynamically emergent from participative practice, and thus part of the open adaptive system of human activity, the computer's ontology will stay much as it is: a closed and rather static system, as it was designed to be.

Near the end of the book, Dourish moves towards design by offering six design principles. To some extent these are used to summarise earlier points, and non-designers may treat this section as a walk through the main points made earlier. However, this section is primarily intended to change design practice. The first principle is *computation is a medium*, which reflects the way that it is not just the external devices for input and output devices that affect a person's use and interpretation of a computer system. The internal digital representations of programs, databases, workflow graphs, files, operating systems and so forth are designed to support some activities and interpretations while inhibiting others. As Dourish emphasises throughout the book, many of the assumptions as to what should be supported and what should be inhibited are hidden or implicit in the craft of system design, and so this principle exhorts designers to be more aware of the communicative significance of their systems—including those not explicitly designed to be tools for collaboration or group work. The second principle points out that this signification occurs in more objective or individual ways, but also as part of intersubjective social practice: *meaning arises on multiple levels*. Here he asks us to design for both of them, as use of a system involves acting on it but also interacting with other people through it. These first two principles should be unproblematic for most readers of this journal, although they suggest a new and broader attitude for most software engineers.

The next two principles may reaffirm to software engineers their unavoidably limited influence on systems' use and interpretation: *users, not designers, create and communicate meaning* and *users, not designers, manage coupling*. The latter refers to the referential coupling between a system's internal representations and the context of use. Given the earlier chapters, these points should be easily absorbed by the reader. With regard to computational media, Dourish does point towards some techniques that go a little way towards supporting these principles, such as macro and visual programming, but he makes clear that there is a long way to go before users can radically adapt and appropriate system structure.

The last two principles are, again, intended to instil a broader design attitude. The fifth principle, *embodied technologies participate in the world they represent*, is phrased in a rather obtuse way, but expresses a point familiar to CSCW: computer systems' representational structures are amongst the artifacts that we work and act with, rather than being outside of work and activity, and merely descriptions of the artifacts of work. To echo a perennial phrase of sociologists in CSCW, we work 'in and through' computational media, and design should take more account of this. The sixth and last principle goes slightly further in defining how embodied technologies participate in the world they represent: *embodied interaction turns action into meaning*. Systems, tools and symbols in computational media are among the utterances, gestures and artifacts-in-use that collectively constitute our activity. Computation is a medium

like any other, and a computational representation is like any symbol in that it gains meaning from its combination in use with each other and with symbols in other more traditional media such as speech, gesture, writing, architecture and so forth. Just like the meaning of the word, the meaning of the system is its use in the language.

Before setting out these principles, Dourish says why he chose this form of influence upon his readers. “Rules would lay down a method for design; guidelines would suggest to a designer what to do. However, given the variety of settings in which the embodied interaction approach is applied, it would be inappropriate to give rules or guidelines here.” I expect that some designers reading the book will feel slightly disappointed with this, as they will be looking for some practical suggestion as to what to do. While it may be true that rules and guidelines can be followed blindly, with little regard to the underlying theories or motivations, a reader who has got this far through the book, and understood Dourish’s arguments, would be unlikely to take such a narrow view. The chapter outlining these principles does offer a number of brief references to systems, but on the whole Dourish holds to the view he states in the concluding chapter:

Embodied interaction is not a technology or a set of rules. It is a perspective on the relationship between people and systems. The questions of how it should be developed, explored and instantiated remain open research problems. (p. 192)

The key or the barrier to these open research problems can be seen near the end of the concluding chapter. Dourish suggests that he has “embraced a nonrepresentationalist stance toward interaction and cognition”. This is true when it is assumed that the representations in question are based on the positivist views of ontology, epistemology and computation, but invites the question of what representations and structures we can say *are* involved in the brain and in language. We should look forward to borrowing design ideas from neurophysiology and evolutionary linguistics—the latter field being quite distinct from and more useful than evolutionary psychology—as well as from designers in HCI and CSCW who use theory to drive and critique systems design.

Looking back over the book, it is clear that it has both style and substance. Dourish does write well, and offers a rich tapestry of argument, example and reference. The book differs from most other texts in building an over-arching understanding of the most important issues facing HCI and CSCW. By bridging between system design and theory, he helps make the more philosophical material accessible to practitioners and also helps make the design issues accessible to readers more familiar with philosophical issues. It should serve as a text for more mature undergraduates, and it should be compulsory for postgraduates in HCI. Most of the common undergraduate HCI textbooks may be more obviously practically-focused, but also appear to be rather ragged patchworks held together by weak conceptual threads. Hopefully, it will be read by many of the designers and developers whose critical awareness Dourish hopes to raise. One would hope that reading it will make them aware of their own practice, making their own work more present-at-hand to them and making explicit many of their normally implicit theoretical assumptions. It does leave open some challenging research questions, especially with regard to the design of deep system structure to support embodied interaction and the interdependence of embodied and non-embodied interaction, but it is a very significant contribution to the literature of HCI and CSCW. *Where the Action Is* is the best book in HCI for years, and everyone who reads this journal should read it. I hope it sells in truckloads.