Managing the Ecology of Interaction

Alan Dix

Computing Department, Lancaster University
Lancaster, LA1 4YR, UK

alan@hcibook.com

http://www.hcibook/com/alan/papers/Tamodea2002/

ABSTRACT

Real work is complex and rich, involving other people, physical artefacts and constant re-planning of tasks to accommodate the contingencies of the situation. This rich ecology of work is often seen as opposed to more structured and formal methods of task analysis and interface design. This paper discusses a range of phenomena related to ecological settings and show how they can be incorporated within formal models. It also discusses several paradigms of interaction that take this into account including the socio-organisational Church–Turing hypothesis and incidental interaction. In conclusion, models can be both rich and also open to unintended uses.

Keywords

situated action, ecological work settings, ubiquitous computing, triggers, artefacts, ethnography, incidental interaction

1. INTRODUCTION

Rationale

Many studies and many areas in HCI and CSCW emphasise the rich ecological setting of human–computer interaction: situated action, distributed cognition, activity theory, ethnomethodological studies. Often those who espouse rich understanding of interaction contrast it with more formal models and reject these as inappropriate for capturing the complexity of human experience and activity.

In the biological disciplines, an appreciation of the complexity of ecological interactions was also a long time coming. However, those studying the biosphere or environmental management do not stop at saying that it is complex, but seek to analyse and model these interactions in order to predict the effects of interventions and manage threatened ecosystems. Similarly, in formal areas of human–computer interaction we need to accept the limitations of our formalisms, but also extend them to incorporate the richer interactions with the work or home environment.

In this context this paper will discuss a variety of issues that can inform or form part of our formal models.

- information – how people use sources to drive and inform their actions
- triggers – understanding why things happen when they happen,
- artefacts – that embody information not just in what they are, but also where they are and how they are disposed in the workplace,
- placeholders – how people remember where they are
- the socio-organisational Church–Turing hypothesis – that organisations perform information processing and thus share features with cognitive and electronic computation;
- incidental interaction – systems that respond because people have acted for some other purpose

These can be seen as part of a broader theoretical perspective of embodied computation – that real computation happens in the physical world and that the physical and human world is full of computation.

By having a rich and, where appropriate, formalised, understanding of the ecology of human–human and human–computer interaction, we are better placed to ensure that our interventions in the workplace do not lead to ecological catastrophes in the organisation.

Overview

In the following section we'll look at some of the background on rich contextual understanding of the workplace and also the evolving nature of interaction. Then in section 3 we will move on to examine some of the phenomena, listed above, that are central to the ecology of work: information, triggers, artefacts and placeholders. The aim is to see how these can be uncorporated as extensions to existing task and user interface models. To give these individual phenomena a broader perspective section 4 will consider a number of paradigms that help us to understand the nature of rich interaction.

Caveat

In discussing the phenomena of a rich ecological view of interaction I will make general statements about the weakness of these aspects within task and process modeling. In fact, all of the phenomena I discuss (except possibly placeholders) are incorporated in some techniques. I acknowledge some of these exceptions, but will inevitable miss out many and apologise to the developers and practitioners of these techniques.

Happily, we live in the world of the web and so this is a living document. On the web page associated with this paper I will have a matrix of task/process analysis and modeling techniques and the phenomena they support. Let me know about your favourite technique.

This said, I do believe that most of the phenomena I discuss are absent in most task modeling notations. Furthermore, it is by considering these phenomena together that we begin to build a richer ecological model of user activity.

2. BACKGROUND

Traditional techniques

Formalised methods such as systems analysis, process modeling and task analysis adopt a systemised, almost Taylorist view of the work place – people working to achieve well-defined goals following regular procedures.

In fact, even the earliest systems analysis texts did take into account the richness of the work environment. I recall reading one text, written in the late 1960's, that described a printshop where productivity was lower than predicted after the installation of new machinery. The analyst was asked to advise on automating the equipment. After observing the workplace he asked for a small budget of a few hundred pounds and the productivity dramatically rose. What did he do? He bought white overalls. The equipment was oily and the operators, mostly young women, were reluctant to work too quickly for fear of damaging their own clothes. The overalls protected their clothes and obviated the need for a computer.

It is perhaps less clear how this sort of factor fits within more recent and more recent formal frameworks.

Rich contexts

Many authors over several decades have criticised formal techniques in HCI on philosophical or empirical grounds. Winograd and Flores [19] criticise formal AI-based cognitive models and instead look to more socially inspired models, in particular Searle's speech-act theory. Strangely this rich view of communication was used to justify the design of Coordinator, arguably one of the most over-formalised tools of all time! Suchman's "Plans and Situated Action" [17] was formative, not just in challenging simplistic models of pre-planned human action (see also later), but also in popularising ethnographic techniques in HCI and CSCW. Her central thesis was that real interaction is not preplanned, but instead acted out in response to the actual work situation.

Since then there have been numerous ethnographic studies all emphasising the incredible richness of human interaction and often the inability of formalised processes to incorporate it. For example, in a study of a printshop (yes another) Bowers et al. [2] found that the operators had to constantly workaround the job management software as it assumed linear patterns of work that did not reflect the contingent and dynamic re-planning necessary on the shopfloor.

More recently Dourish [9] drew on this broad tradition and the phenomenological roots in philosophers such as Heidegger and Wittgenstein. We will return to his design methods for embodied interaction later.

In a philosophically different strand of work the distributed cognition literature has challenged the model of cognition "in the head" and instead suggests that real cognition happens in interaction with the environment and with each other. One classic study showed how Polynesian sailors were able to navigate without formal charts and without the requisite experience in any individual's head [12].

All of these authors and studies emphasis the richness of real interaction. We operate within an ecology of people, physical artefacts and electronic systems.

Rich interaction

This rich ecology has recently become more complex as electronic devices invade the workplace and our day to day lives. Are methods designed to deal with traditional office information systems able to adapt to incorporate mobile systems, wearable computers and ubiquitous applications?

3. PHENOMENA

Situatedness – in dialogue with the environment

One could say that the lesson of situated action and of distributed cognition is about the parity in relationship between the 'actor' and the world. We do not just *act on* the world, but *act with* the world. We are driven by what we see and hear from other people, from automated systems and from the physical objects in the world. In response our actions, words and sometimes gestures and demeanor speak back into that rich world.

In day to day life we understand about dialogue with other people. In HCI we are used to thinking about dialogue between users and the computer system. However, in a full ecological analysis we must also accept that users are in dialogue with the physical environment. We use the information stored in artefacts and their physical disposition to trigger and guide our actions and the physical properties of the world limits and constrains our actions on it.

In the rest of this section we will look at several phenomena of this dialogue with the environment.

Collaboration – doing it togather

Many notations and methods already handle collaboration explicitly [3,14,16]. There are two types of model. One is where the process as a whole is mapped out and parts assigned to each person (common in function allocation [3], workflow and process methods). The other is where several role-oriented models interact as is the case with CTT [16]. These are complementary representations and can be handled together with suitable tool support.

It is interesting to note that the roles identified typically include both humans and automated systems, but rarely aspects of the physical environment. However, it is only a small step to imagine treating the environment or parts of it as dialogue partners alongside the human and computer.

Information – what you need to know and when you need to know it

I recall when writing the first edition of our HCI textbook [7], we bemoaned the fact that cognitive models took an almost totally output and action dominated view of human cognition. We have goals, which translate into sub-goals and so on until we perform actions – an entirely head-outward flow of control. In a similar vein, Suchman's "Plans and Situated Action" was particularly critical of the AI inspired views of human planning, which were again largely based on creating internal plans based on internal models of the world, which are then 'blindly' executed. I use the word 'blindly' here quite carefully, as these are indeed sense-less models of human action.

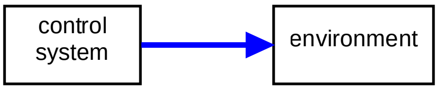


Figure 1. open loop control

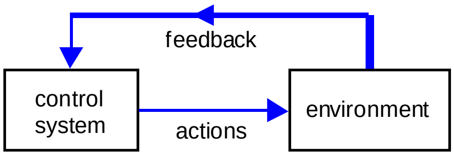


Figure 2. closed loop control

In control engineering these output-only models would be described as open-loop control (figure 1) as opposed to closed-loop control (figure 2), which constantly monitors the effects of its outputs on the environment and uses these to modify future behaviour.

In general closed-loop control is more robust and it is not surprising that both internal physiological processes and external human behaviour are typically closed-loop systems. Indeed, the user interface literature is full of the importance of feedback and effective information display; it is just that the early formal models have often left this out.

There are several examples of cognitive models that do take this feedback loop seriously. D-TAG is a display-based version of task action grammar [10] and there have been several other variants of display-based models. Interacting cognitive subsystems (ICS) is focused strongly on the transformations of representation during the perception-to-action cognitive cycle [1]. Also the earliest papers on cognitive complexity theory (CCT) included perceptual operators on the production-rule-based cognitive model component, but strangely it was the actions only that were matched against the system dialogue model [13].

It is not uncommon to see references to information seeking in the names of tasks in task models, but this is normally where the information seeking activity is regarded as a substantive task. In practice, information is used throughout task execution. For example, in the simple tea making task (figure 3), the "boil kettle" subtask does not require any information, but the "get out cups" task requires the actor to know how many are required. Does he remember, or does it need to be written down?

Information is central to several task analysis methods, such as TAKD [4], however these are focused on what kinds of things the user needs to know in general – ontology and domain modeling – not on what the user needs to know at a particular moment.

It is a simple matter to add an information analysis stage to any task analysis method or notation. Note that some tasks have no information requirements – other than the fact that they are to happen. For example, the "make pot of tea" subtasks requires no information other than the fact that the kettle has boiled. However information is required whenever:

(a) a sub task involves inputting (or outputting) information
(b) there is some kind of choice
(c) a subtask is repeated a number of times that is not prespecified

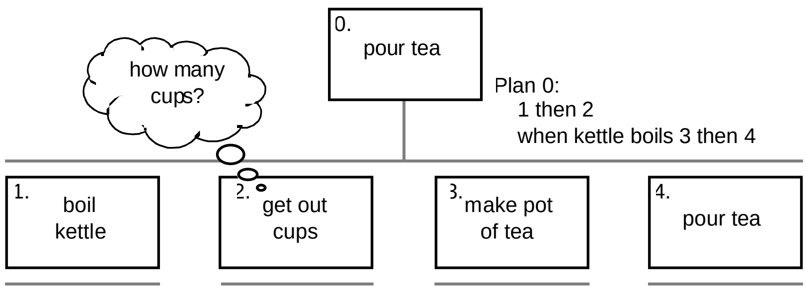


Figure 3. tea making task

Note that (c) is a special case of (b). To detect (a) one needs to look at the kind of task, whereas (b) and (c) are evident from the temporal structure of the task (for example, in the case of HTA, this would be in the plan).

Having discovered that information is required it may come from several sources:

(i) It is part of the task (e.g., in the case of a phone call, who one is going to phone)

(ii) The user remembers it (e.g., remembering the number after ringing directory enquiries)

(iii) It is on a computer/device display (e.g., using a PDA address book and then dialing the number)

(iv) It is in the environment either pre-existing (e.g., number on phone directory) or created as part of the task (e.g., number written on piece of paper)

Reducing memory load is part of standard usability guidelines. Knowing what information is required during a task allows us to design or redesign the task so that information is available when required. An infamous example of this are those all too common modal dialogue boxes that ask you some question but hide the window that has the information you need to answer the question!

In most multi-windowed GUIs it has been possible for user interface designers to be quite careless about information requirements. One can make so much information available and let the user layout different windows to perform the task. In contrast industrial control design is far more careful about knowing what is required as there are often very many possible values to display, but the operators may have very little time to respond to an alarm and so cannot browse complex menu systems to find information. As user interaction moves away from the computer screen to dedicated devices, WAP phones, interactive television screens and smart appliances these issues of careful information requirements analysis will become significant for all applications.

Triggers – why things happen when they happen

Workflows and process diagrams decompose processes into smaller activities and then give the order between them. Similarly in HTA plans give some specification of the order of sub-tasks and in CTT these temporal orders are made more specific using operators derived from LOTOS.

Figure 4. simple work process

Figure 4 shows a simple example, perhaps the normal pattern of activity for an office worker dealing with daily post. Notice the simple dependency that the post must be collected from the pigeonhole before it can be brought to the desk and before it can be opened. However, look again at the activity "open post" – when does it actually happen? The work process says it doesn't happen before the "bring post to desk" activity is complete, but does it happen straight away after this or some time later?

In previous work with Devina Ramduny and Julie Wilkinson [5,6], we have looked in detail at the triggers that cause activities to happen when they happen. In the case of opening post this could easily be something like "at coffee time" rather than straight away. In our work we identified a number of common triggers:

- *immediate*: straight after previous task
- *temporal*: at a particular time or after a particular delay
- *sporadic*: when someone things about it
- *external event*: some event occurs such as a phone call
- *environmental cue*: something in the environment prompts action

We can augment the work process with triggers for each activity (figure 5).

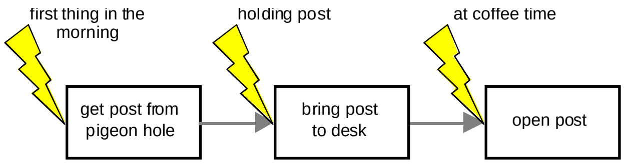


Figure 5. triggers for activities

Notice how we have examples of several types of trigger, two temporal and one environmental (letters in the office workers hand prompting her to carry them to her desk).

Triggers are important not only for understanding the temporal behaviour of the task, but also because they tell us about potential failure modes. If two environmental triggers are similar one might do parts of the task out of sequence, if a trigger may not occur or be missed (likely for sporadic triggers) activities may be omitted entirely. Triggers also help us assess the likelihood of problems due to interruptions – for example, immediate "just after" sequences are disrupted badly but environmental cues tend to be robust (because they are still there).

Sometimes triggers are seen in the plans of HTAs and sometimes 'waiting' subtasks are included for external events, but these are both the exception and the normal assumption is that tasks are uninterrupted. However, it is straightforward to add a trigger analysis stage to most task analysis methods.

In terms of the ecology of interaction, triggers remind us that tasks are not performed Magnus Magnusson fashion "I've started so I'll finish". In practice, tasks are interleaved with other unrelated tasks or, potentially more confusing, different instances of the same tasks and may be interrupted and disrupted by other activities and events. Furthermore the performance of the tasks is dependent on a host of, sometimes fragile, interactions with the environment and apparently unconnected events.

Artefacts – things we act on and act with

Notice that one of the trigger types is environmental cues things in the environment that prompt us to action. Some years ago I got a telephone call reminding me to respond to a letter. I couldn't recall receiving it at all, but searching through a pile on my desk I found it and several other letters over a period of several weeks unopened and unread. What had happened? My practice was to bring the post upstairs to my desk, but not always read it straightaway. Not being a coffee drinker it was not coffee time that prompted me to open the post but just the fact that there was unopened post lying on my desk. This process had worked perfectly well until there was a new office cleaner. The new cleaner didn't move things around on my desk, but did 'tidy': straightening up higgledy-piggledy piles of paper. However, I had unconsciously been using the fact that the bundle of unopened post was not straight as a reminder that it needed dealing with. So post that for some reason got missed one day would then look as if it was tidily 'filed' in a pile on my desk.

This story is not unique. The ethnographic literature is full of accounts of artefacts being used to manage personal work and mediating collaborative work. Some of that purpose is to do with the content of the artefacts – what is written on the paper, but much by the physical disposition – by orienting a piece of paper towards you I say 'please read it'. In the case of my desk the cue that said "post needs to be opened" was purely in the physical orientation (not even the position).

Of course, artefacts do carry information and are often the inputs or products of intellectual work. Furthermore, in physical processes the transformation of artefacts is the purpose of work.

One example that has been studied in detail in the ethnographic literature is air traffic control and all these uses of artefacts are apparent [11]. Flight strips are central (figure 6) – small slips of card for each aircraft recording information about the aircraft (flight number, current height, heading etc.). this information is important both for the controller managing the aircraft, but are also an at-a-glance representation of the state of the airspace for other controllers. However, the controllers also slightly pull out strips corresponding to aircraft that have some issue or problem. This acts partly as a reminder and partly as an implicit communication with nearby controllers. Finally, the strips in some way represent the aircraft for the controllers, but of course, the real purpose of the process is the movement of the aircraft themselves.

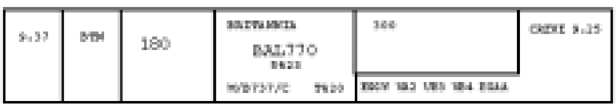


Figure 6. air traffic control flight strip

Task models often talk about objects, either implicitly in the description of subtasks or explicitly in the task model. However, the objects are always 'second class' – users act on them, but they are not 'part of' the task. CTT and most work process notations do talk about automated tasks, but not the artefacts, whether electronic or physical included within the interaction.

In UML and other OO design method it is common to give lifecycle description of 'objects', however, these is usually because we are intending to store and automate the object electronically. Also workflow analysts study document lifecycles – again largely because of the intention to automate.

In the task analysis chapter of our HCI textbook, my co-authors and I do treat physical objects as 'first class' within an example of entity-relationship style task analysis. This was based largely on the ATOM method [18], but, to my knowledge, this style of method has not gained widespread acceptance.

There is no reason why most task analysis methods should not adopt some form of artefact tracking. This may be as simple as recording which artefacts are triggers for, used by, modified by, or produced by any particular sub-task. For tasks where artefacts are particularly central more sophisticated artefact lifecycles could sit alongside the task description. These lifecycles may be mundane (letter closed → letter open), but this is the point, users recruit their everyday knowledge and physical properties of the world to coordinate their activity.

Placeholders – knowing what happens next

It is half past five in the evening. The busy office building is beginning to quiet as people pack up to go home. One or two work late in their offices, but as the evening wears on they too go home. Soon there is only the hum of vacuum cleaners and the clatter of wastebins as the office cleaners do their work, until eventually, the last light goes out and the building sleeps. A few have taken papers and laptops home and continue to work, but eventually they too put aside their work and sleep.

It is three o'clock in the morning, in the darkness and silence of the office and the deep sleep of all the employees, where is the memory of the organisation? The next morning at nine o'clock the office is a flurry of activity, it has not forgotten and has restarted its activities, but how?

We have already discussed two aspects of this memory: information required to perform tasks, and triggers that remind us that something needs to happen. However, there is one last pieces of this puzzle that we have hinted at several times already. As well as knowing *that* we need to do something we need to know *what* to do next. In the complex web of tasks and subtasks that comprise our job – *where* are we?

In fact, when looking at triggers we have already seen examples of this. The post being untidy on my desk said both "something needs to happen", but the fact that it was also unopened said, "it needs to be opened". In that discussion we already noted that similar triggers could cause sub-tasks to be performed out of sequence. If we only have a small number of dissimilar tasks this is unlikely to happen as we can remember where we are in

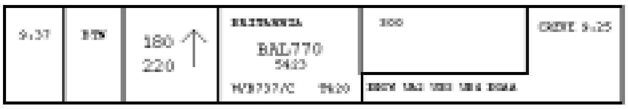
Figure 7. flight level management task

each task. However as the number of tasks increases, especially if we are performing the same task on different things, it becomes harder to remember where we are.

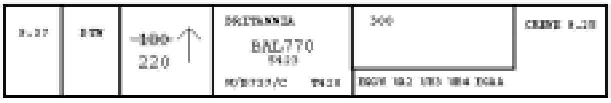
Let's look again at air traffic control. One of the controller's tasks is to manage the flight level of aircraft. A much-simplified model of this activity is shown in figure 7. Because this is a shared task between the controller and the pilot, each box is labeled with the main actor (although tasks 2 and 3 are both communications). Recalling earlier sections we might ask what information is required at each stage, for example task 1 would depend on radar, locations of other planes, planned take-off and landings, new planes expected to enter airspace.

Note that box 5 is not really a task more a 'state of the world' that signifies task completion, however, it is important as the controller will need take alternative actions if it doesn't happen. Of course, without appropriate placeholders the controller might forget that a plane has not achieved its target level either causing trouble later as the old level will not be clear or even potential conflicts between aircraft.

In fact, the flight strips do encode just such a placeholder (see figure 8). When the controller informs the pilot of the new height he writes the new level on the flight strip (i). When the pilot confirms she has understood the request the pilot crosses out the old level (ii). Finally when the new level has actually been reached the new level is ticked (iii).



(i) controller gives instruction to pilot "ascend to flight level 220"



(ii) pilot acknowledges the instruction

(iii) new height is attained

Figure 8. flight strip annotated during task

Virtually all task-modeling notations treat the placeholder as implicit. The sequence of actions is recorded, but not why the user should do things in the way proposed. Of course, one purpose of task analysis has been to produce training – that is to help people learn what appropriate processes are, but this doesn't help to actually remember where you are.

Just like other forms of information, placeholders may be stored in different ways:

(a) in peoples' heads – remembering what to do next

(b) explicitly in the environment – to-do-lists, planning charts, flight-strips, workflow system

(c) implicitly in the environment – is the letter open yet?

Although often forgotten, placeholders are crucial in ensuring that tasks are carried out effectively and in full. At a fine scale it is rare to find explicit records as the overhead would be too high. Instead (a) and (c) predominate. As users' memory may be unreliable when faced with multiple tasks and interruptions, it is not surprising to find that various forms of environmental cue are common in the workplace. However, electronic environments do not have the same affordances to allow informal annotations or fine 'tweaking' of artefacts disposition.

The intentional cycle

In the extreme, when faced with multiple complex tasks in a disruptive environment, we begin to behave in a stimulus–response mode triggered by the environment to act and using the information around us in physical and computational artefacts to decide what to do next. The various phenomena we have considered together make up the aspects of the environment that drive us in this intentional cycle (figure 9).

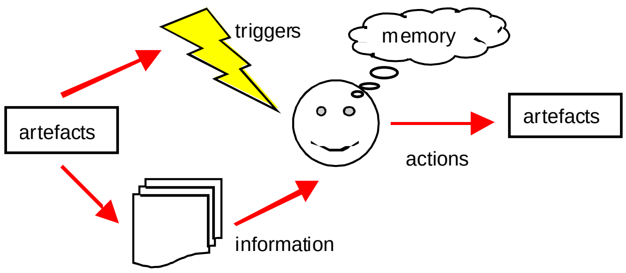


Figure 9. stimulus–response interactions

We can use a more structured and ordered task model to say what we would like to happen, but this is likely to be 'implemented' in the workplace by a more stimulus–driven behaviour.

PARADIGMS

We have seen a number of individual phenomena and how they may be incorporated into richer task models. In this part of the paper we'll examine a few broader paradigms on contextual interactions which can give a framework to the understanding of users' activities in a rich environment.

	computer	human cognition	organisation
process	program	procedural memory	processes, tasks
data	data	long-term memory	files
placeholder	program counter	short-term memory, activation	human memory, disposition of artefacts
initiative	interrupts, event-driven programs	stimuli	triggers

Table 1 - computational parallels

Socio-organisational Church–Turing hypothesis

An organisation has many facets: social, political economic, but amongst these virtually all modern organisations perform some form of information processing … or in other words computation.

The Church–Turing theorem proved that Church's mathematically-based lambda calculus and Turing's more mechanical machine were identical in computational ability. Extrapolating from that the Church–Turing thesis says that all forms of computation are effectively equivalent, and in the sixty plus years since this has not been refuted – even quantum computing merely speeds up computation.

Furthermore, there is a remarkable similarity of structure between many computational devices, partly driven no doubt by conservatism of design. This structural similarity has been used very effectively within cognitive science regarding cognition as having some of the same properties as mechanical computation.

In the same way we can see parallels between computation and organisational information processing. The *socio-organisational Church-Turing hypothesis* is precisely that, the assumption that organisations are likely to exhibit similar structures and phenomena to electronic computers. Table 1 shows parallels between computers, cognition and organisations. In a way a multi-user task model is a kind of 'program' for the organisation and placeholders are program counters. Because the organisation is doing several things at once, like a multi-processor or multi-threaded program it needs many placeholders (program counters) for each task (program thread).

Embodied computation

As well as learning about organisations I have found that this parallel helps in understanding computation itself. We tend to think of computation as an abstract thing, but it was born out of the dual aims to formalise human reasoning (Hilbert, Whitehead, Russell, Church) and to model mechanical algorithms (Turing). However, seeing computation in organisations and also ubiquitous devices reminds us that computation is very physical and I call the recognition of this physicality *embodied computation*. Because computation is achieved in the physical world there can only be a finite amount of memory and finite amount of computational power in any finite space. The Turing machine embodies this with a finite computational engine travelling over a one-dimensional tape storing a finite amount of information in each unit of physical tape. This means that computation is intimately connected to the topology of space and communication, and is typically implemented by cooperative behaviour and transformations of representation – just like human work.

Embodied interaction

Paul Dourish uses a similar phrase but with a very different meaning in his book "where the action is" [9]. Dourish is focused particularly on tangible computing and social computing and he talks about designing for *embodied interaction*, which is interaction where the users drive the meaning and the objects of interaction are often intimately connected to the physical world. His design principles are all connected with giving users the means to use the computer to serve their own adapting activity. This is based on extensive research showing that users constantly adapt and coerce systems designed for one purpose to meet their evolving and discovered requirements. Dourish says accept this and design for it.

This runs counter to the more pre-planned world of most task modeling, but is far closer to the more stimulus-driven model. However, the two paradigms can be complementary. Given a structured task model we can are design systems limited to the pre-ordained sequences. Alternatively we can use the same model not to constrain but to drive the development of systems rich enough to cover them and other, yet unforeseen, tasks and goals.

Implicit interaction

More radical still is *incidental interaction* [8]. Most traditional applications are assumed to be purposeful – the user wants to achieve something and uses the system to do so. This is even true of more 'situated' interactions where the goal is formed as part of interaction. However, many ubiquitous computing applications have a differently character. You walk into a room and the air conditioning system detecting your presence adjusts the thermostat to your normal preference. Your *purpose* is to go into the room, and *incidentally* the thermostat is altered.

Many intelligent and adaptive interfaces within the electronic environment also have this incidental character. For example, while shopping on the Amazon web site the system *incidentally* builds a model of your preferences and thus offer you related books to buy. Another example, with both physical and computational aspects, is the Pepys system [15]. Each employee of Xerox' Cambridge laboratories had an active badge which was tracked using sensors in the offices. At the end of the day Pepys gave everyone an electronic diary of the day based on where they had been and who they had 'met'.

Sometimes these incidental interactions perform a completely separate activity to the user's main task, sometimes they contribute to it, perhaps allowing the system to tailor itself to the user. For example, when you get into the car the interior lights automatically come on, contributing to the task but not initiated by purposeful user action.

These interactions are often probabilistic both in that they may be driven by unreliable environmental sensors (two people went into the room, and only one left, but which one?), statistical or neural inference techniques, and sometimes an attempt to infer the user's current task.

Combining these with traditional task analysis may be a greater challenge than simply 'adding' environmental phenomena. One way is to add incidental interaction as an unreliable 'artefact' lifecycle. Another is to focus on the 'goal' of the incidental interaction itself and regard the users as having 'lifecycles' (go in room → be in room → leave room)

Dealing with such issues is not just important for ubiquitous applications, but is germane to many collaborative interactions where overhearing and peripheral awareness are central to coordination. This suggests being able to deal with loosely coupled, uncertain interactions is a major future challenge for task modeling.

5. DISCUSSION

Real work and life is a rich, situated, contextual thing. However, technological design by its nature embodies implicit models of activity and those from more contextual design traditions who claim to eschew such formalisation are being disingenuous. We have seen how elements of ecological work environments can be included explicitly within formal modeling frameworks. This is not to say we expect to capture the full richness of human activity, to think this would be foolish. Our formal models must both embody rich representations of ecological phenomena and still be open to unexpected uses.

ACKNOWLEDGEMENTS

This paper is based particularly on work over several years with Devina Ramduny and Julie Wilkinson and also work related to EPSRC funded projects CASCO and DIRC.

REFERENCES

1. Barnard, P. and J. May (1995) Interactions with Advanced Graphical Interfaces and the Deployment of Latent Human Knowledge. In *Proc. of Design, Specification and Verification of Interactive Systems, DSVIS'94* . F. Paternó (ed). Springer.
2. Bowers, J., G. Button and W. Sharrock (1995). Workflow from within and without: Technology and cooperative work on the print industry shop floor. In *Proc. of ECSCW'95*, Kluwer.. pp 51–66.
3. Dearden, A., M. Harrison and P. Wright (2000). Allocation of function: scenarios, context and the economics of effort. *Int.l J. of Human-Computer Studies*, **52**(2):289-318.
4. Diaper, D. (1989). Task Analysis for Knowledge Descriptions (TAKD); the method and an example. In *Task Analysis for Human-Computer Interaction*. D. Diaper (ed.), chapter 4, pp. 108-159. Ellis Horwood.
5. Dix, A., D. Ramduny and J. Wilkinson (1998). Interaction in the Large. *Interacting with Computers*. **11**(1): 9-32.
6. Dix, A., J. Wilkinson and D. Ramduny (1998). Redefining Organisational Memory - artefacts, and the distribution and coordination of work. In *Understanding work and designing artefacts* (York, 21st Sept., 1998)
7. Dix, A., J. Finlay, G. Abowd and R. Beale (1998). *Human–Computer Interaction* (2nd ed). Prentice Hall,.
8. Dix, A. (2002). *Incidental Interaction*. available online: http://www.hcibook.com/alan/topics/incidental/
9. Dourish, P. (2001). *Where the Action Is: The Foundations of Embodied Interaction*. MIT Press.
10. Howes, A., and S. Payne (1990). Display-based competence: towards user models for menu-driven interfaces. *Int. J. of Man-Machine Studies*, **33**:637-655.
11. Hughes, J., J. O'Brien, M. Rouncefield, I. Sommerville and T. Rodden (1995). Presenting ethnography in the requirements process. In *Proc. IEEE Conf. on Requirements Engineering, RE'95*. IEEE Press, pp. 27–34.
12. Hutchins, E. (1990), The Technology of team navigation. In *Intellectual teamwork: social and technical bases of collaborative work.* Gallagher, J., Kraut, R. and Egido, C., (eds.), Lawrence Erlbaum.
13. Kieras, D. and P. Polson (1985). An approach to the formal analysis of user complexity. *Int.l J. of Man-Machine Studies*, **22**:365-394
14. Limbourg, Q., C. Pribeanu and J. Vanderdonckt (2001). Towards Uniformed Task Models in a Model-Based Approach. *In Proc. DSV-IS 2001*. Springer pp.164-182
15. Newman, W., M. Eldridge and M. Lamming (1991). Pepys: Generating Autobiographies by Automatic Tracking. In *Proc. of ECSCW '91*. Kluwer... pp 175-188.
16. Paternó, F. (1999). *Model-based design and evaluation of interactive applications*. Springer.
17. Suchman, L. (1987). *Plans and Situated Actions: The problem of human–machine communication*. Cambridge University Press,.
18. Walsh, P. (1989). Analysis for task object modelling (ATOM): towards a method of integrating task analysis with Jackson system development for user interface software design. In *Task Analysis for Human-Computer Interaction*. D. Diaper (ed.), pp. 186-209. Ellis Horwood.
19. Winograd, T. and F. Flores (1986). *Understanding computers and cognition : a new foundation for design.* Addison-Wesley.