beyond intention
pushing boundaries with incidental interaction

Alan Dix

Lancaster University, Lancaster UK
alan@hcibook.com
http://www.hcibook.com/alan/topics/incidental/

Abstract. Traditional interaction is purposeful, but interactions based on sensing and environmental context are of lower salience and require less explicit attention. The extreme end of this spectrum, but becoming more common, is implicit interaction, which occurs when a user action intended for one purpose is interpreted to achieve some other goal. Because it is an extreme point, incidental interaction acts as a touchstone allowing us to explore the limits of traditional interaction models, task analysis, cognition and software architectures. We find that all either break down or require rethinking in the face of these radically novel interactions.

1. introduction

In traditional computer applications a user was expected to approach the system with a clear intention to perform some activity or achieve some goal. The actions were purposeful and direct and the results were explicitly attended to and evaluated. The design emphasis is on making the affordances of interaction unambiguous and available and ensuring that system feedback and state are clearly visible.

However, in many areas of human computer interaction we have seen a growing number of systems and interaction paradigms where user attention and intention is lower. In CSCW the concept of awareness has been central for many years and similarly ambient interfaces emphasise low salience displays of background information. Paradigms such as calm interfaces, tacit and implicit interaction all emphasis output that is non-intrusive and ecologically natural forms of input/control.

Whereas the traditional interface was based around controls and input devices, these low attention and natural input paradigms are more closely related to sensing technologies and contextual interpretations.

At the extreme end of the spectrum are interactions where actors (users) perform actions for some purpose (say opening a door to enter a room) and the system senses this and uses it for some purpose of which the actors are unaware (perhaps adjusting the air conditioning), but which affects their future interactions with the environment or system. I call this *incidental interaction*.

In this paper we'll look briefly at the definition of incidental interaction and some examples. We'll then see how this challenges many areas of traditional interaction design: the fundamental execution–evaluation cycle implicit in much of HCI, the notion of task and task analysis, the limits of cognitive abilities and the impact of software architecture. Incidental interaction is being used to highlight these issues, but it is evident that these are problematic in many forms of low attention and low intention interaction.

2. incidental interaction

definition

where actions performed for some other purpose, or unconscious signs,
are interpreted in order to influence/improve/facilitate
the actors' future interaction or day-to-day life

The above definition is expanded in detail on the incidental interaction web pages (above), but we will briefly review the main issues.

The actor has some purpose other than that of the actual incidental interaction (although the results may impact on the user's activity) and may not even be consciously doing anything at all. As an example of the latter the Commonwealth baton had an electronic 'flame' that flashed depending on the bearer's heart rate.

The system or application interprets these explicitly. We are not talking about unintended side effects. Although incidental interactions are not intentional from the actor's perspective they are intended by the designer.

This interpretation is then used to in some way improve or facilitate the actor's future interactions with the system (e.g. adapting the interface) or the actor's environment (e.g. adjusting the heating).

Finally note that the definition is phrased in terms of the plural 'actors'. This is to deliberately include cases where the property being sensed is related to a group or relationship (e.g. a meeting), and also to include those where the effect of the incidental interaction is to influence some other member of the group rather than those directly sensed (e.g. some awareness mechanisms).

We can position incidental interaction at one end of a continuum:

intentional	press light switch
expected	walk into room expecting lights to switch on
incidental	walk into room and unbeknown to you air conditioning increases

examples

We have already seen some examples in the description above. The automatic lights are quite interesting as a small change would turn them from an expected interaction to an incidental one. Imagine a house had real light switches, but (for energy saving) always switches off the lights in rooms that no-one is in. When you approach the room the lights automatically turn on, so that you are never aware that they switch off. Doors that automatically unlock when an active badge is near are another example.

Car courtesy lights have more complex behaviour. Depending on the model they may turn on when the doors are unlocked or when the doors are opened. They may turn off after some fixed time, when the doors are closed or when the engine is started. Underlying this is some designer's model of the task of getting into a car – perhaps sorting out belongings in the car, looking at a map, etc., before setting off. The sensors are unreliable means of detecting the user's intentions, but the incidental interactions with the lights are designed to support the task. Note, the driver's *purpose* is to get into the car and *incidentally* the lights come on.

In the Pepys project at Xerox EuroPARC , all staff wore an 'active badge' that detected their location in the building using infra-red sensors [[WHFG92]]. The Pepys system analysed the logs of people's location at the end of each day and used these to produce a personal diary for each person [[NEL91]]. Because Pepys knew about the layout of the offices, and who was where when, it was able to detect when two or more people were in the same location and create a diary entry "had meeting with Allan and Victoria". Note again Allan and Victoria's *purpose* is to visit Paul's office and *incidentally* a diary entry is created for each of them.

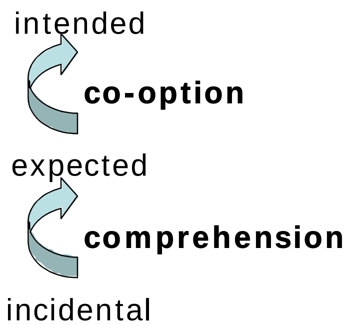
The MediaCup [[GBK99,BGS01]] also facilitates incidental interaction. It is a base unit that can be attached to an ordinary coffee mug and detects movement (when drinking, walking around, etc.), pressure (for fullness), temperature (fresh coffee vs. old) and location. The information gathered by this gives some indication of the drinker's current activity and location, which can then be used for community awareness. Hans' *purpose* in filling the mug is to have a drink of coffee and *incidentally* his colleagues become aware he is taking a break.

Incidental interactions can also take place entirely within the electronic domain.

In many electronic shopping sites the system keeps track of the items you have purchased or examined and then suggests additional products based on your inferred tastes. Your *purpose* is to buy product X and *incidentally* the system infers your tastes and suggests product Y.

One system I have worked on personally was onCue. This is an 'intelligent' toolbar that sits on the side of the user's screen. When the user cuts or copies any text onCue examines the clipboard. It analyses the content and depending on the type of material changes the tool items to reflect it. If the copied text is a post code, onCue suggests internet mapping services; if it looks like a name (initial capitals etc.), onCue suggests internet directory services; if the text is a list or table of numbers, onCue suggests spreadsheet or graphing applications. As well as the triggering event being implicit, the 'suggestion' is deliberately low salience, the currently suggested services slowly fade in as small icons in the toolbar. Note again, the user's *purpose* is to copy the text somewhere else and *incidentally* potential useful services get offered.

fluidity



The continuum from intentional, through expected, to incidental interaction is largely about purpose – what the user thinks – not the actual system itself. Of course, certain types of system will suggest more or less strongly one mode of interaction, but there is some fluidity depending on the user's experience, awareness, etc. As the users become more aware of the interactions happening around them they may move through the continuum towards more purposeful interaction.

- **comprehension**: incidental → expected – If users notice, consciously or unconsciously, that incidental interactions are taking place they may come to expect or rely upon them. For example, if I realise that the courtesy lights come on when I get into the car I may leave checking my route until I get in knowing that it will be lit.
- **co-option**: expected → intended – When I know that something will happen when I perform an action, then I may deliberately perform the action to cause the effect. For example, I may deliberately open and close the car door to trigger the courtesy light mechanism.

3. challenging our models

As well as being an interesting interaction paradigm in its own right, incidental interaction really pushes our fundamental assumptions about interaction and our ways of modelling it. I will not try and 'solve' these issues within this paper, but instead will raise the problematic aspects as a challenge for future HCI theory and practice.

interaction models

The explicit or implicit model behind nearly all interaction is some form of intentional cycle such as the Norman execution–evaluation loop [[N90]]. The user has some goal (intention), formulates some action that he/she believes will act towards that goal, performs the action and then re-formulates future actions based on the feedback.

In traditional cognitive modelling this was seen as very plan driven with goal stacks and hierarchies etc. In these accounts, the intentional cycle is seen as starting with the user, even to the point that the effects of feedback are often ignored. In more contextual accounts of interaction such as situated action [[S87]] or distributed cognition [[H90]], the goals or intentions are more at the level of overall motivations or end-state aspirations and the focus tends to shift to a cycle of activity starting with the state of the world and recent system 'responses' with the user acting on the world in response to the current state. However, this is still clearly purposeful activity.

Incidental interaction and to a lesser extent expected interactions do not fit this picture. The user and system share the experience of the user's actions, but the purposeful activity of the user is distinct from the intended outcomes of the system.

This is not just a theoretical issue as it is an underlying assumption that cuts through nearly every usability guideline, principle and method. For example, the importance of rapid and visible feedback is based on the assumption that users need to understand fully the effect of their actions. In incidental interaction and low awareness applications the opposite is often true.

task analysis

Traditional task analysis is also highly purposeful, although debatably less wedded to this than cognitive models. Certainly to cope with more contextual interactions task analysis methods need to evolve to include or link to representations that are more about the physical world and the rich ecology of lived work or domestic life. In a previous paper I have examined some of the potential issues and extensions that may be necessary for this [[D02]].

However, incidental interaction poses a more fundamental question – what task do we need to model? In incidental interaction we have two 'tasks' that are occurring.

(i) the user's primary task – their purposeful activity.

(ii) the task that the incidental interaction is attempting to support or achieve

Often, as in the case of the courtesy light, the two are the same task, but it is used in different ways. The user's purposeful activity is assumed to occur, to a large extent, independently of the system's actions. We need to model it in order to computationally interpret the user's actions as activity. In contrast, we need to model (ii) in order to understand how to facilitate or progress it.

cognition

Natural *inanimate* physical things have a set of properties intrinsic to their physicality:

- directness of effect – You push something a little and it moves a little, you push hard and it moves a lot.
- locality of effect – Effects are here and now. If you push a rock and then 2 seconds later it moves you would be disturbed.
- visibility of state – Solid objects have a small number of relevant parameters that define their dynamic state: location, orientation. We have some difficulty with invisible properties such as velocity and even more when physical things do have hidden state, for example, the 'joke' balls that have a ball bearing inside and so do not move in a straight line. Of course this example is not natural but constructed.

We have evolved as creatures to cope with physical things and other creatures, not technological devices. Although we have higher level reasoning that enables us to cope – the same reasoning that enables us to create technology, this is only significant when we 'think about' things, our more innate cognitive abilities are shaped by the natural.

Computer systems (and other complex technology such as electrical and pneumatic) break these intrinsics of physicality. Computation creates complexity of effect, networks introduce non-locality in space, memory non-locality in time and a computer has a vast number of invisible variables in its hidden internal state.

We cope (just) with this because either we rationalise and use higher-level thinking to make sense and to make models of these complex non-physical interactions, or we treat the computer as animate. In addition, one of the reasons for the development of the GUI interface style is that it makes the electronic world more like real (inanimate) things.

In incidental, expected and low awareness interactions the intention is that the user is not paying attention to or is unaware of the system's activities. That is, we are not able to bring our higher cognitive abilities to bear and are dependent on our more innate intelligences, which are of course ill prepared for unphysicality. To make matters worse, the system activities are often triggered by physical actions and movements of the user and are manifest in the physical world. In a computer system we are able to re-frame ourselves to expect the odd or the apparently magical actions to occur. In the real world these are deeply disturbing.

system architecture

In incidental interactions it is very likely that sensors are not used solely for their original purpose. This suggests the need for quite open architectures. For example, in designing onCue we used a very open blackboard-style architecture for exchanging information between self-discovering and self-configuring components [[DBW00]]. Unfortunately, when we get down to the level of individual applications it is far harder to get contextual information without writing special code for each potential application. This is one of the reasons for using copy/cut to the clipboard as the main trigger – it is one of the few cross-application standards.

Furthermore, many of the contextual interactions envisaged in this area occur in domestic or other private environments. If we are not careful architectural openness could violate privacy – imagine if the can of beans (with intelligent food label) you just bought communicated back to the manufacturer the contents of your food cupboard.

Highly contextual interactions must also take on board the fact that many of the most important phenomena are not events (things that happen at specific moments), but status (things that always have some measurable value). My own work on status–event analysis has highlighted common phenomena that can be used to understand such systems, but this also impacts on the underlying system architectures [[D91,DA96]].

4. so what?

Probably the dominant type of context-sensitive interaction is still where the environmental context influences the purposeful activity. In contrast, in incidental interaction, the context of the user's current purposeful interaction is used to drive a secondary interaction. This is perhaps context driven not just context sensitive.

The extreme nature of incidental interaction 'breaks' many traditional conceptual, analytic and implementation models. However, these are not just problems for designing incidental interactions. To some extent incidental interaction is highlighting contradictions that apply to other areas of contextual and low-attention interaction styles.

So, as well as being an important class of interactions to study in its own right, incidental interaction can function as a touchstone for those creating new frameworks, architectures and models for sensors, context-sensitive applications and ubicomp design frameworks.

I am afraid this paper presents few answers, more questions and suggestions about the future. But I hope it also begins to set a framework to guide the process of answering these questions.

5. acknowledgements

This paper draws on discussions with and inspiration from many colleagues and especially work related to the EPSRC funded CASCO, EQUATOR, and DIRC projects.

6. references

[[BGS01]] Beigl, M., Gellersen, H.-W. and Schmidt, A. MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects. Computer Networks, Vol. 35, No. 4, Special Issue on Pervasive Computing, Elsevier, March 2001, pp. 401-409.

[[D91]] A. J. Dix (1991). Status and events: static and dynamic properties of interactive systems. Proceedings of the Eurographics Seminar: Formal Methods in Computer Graphics, Ed. D. A. Duce. Marina di Carrara, Italy.

[[DA96]] A. Dix and G. Abowd (1996). Modelling status and event behaviour of interactive systems. Software Engineering Journal, 11(6) pp. 334-346.

[[DBW00]] A. Dix, R. Beale and A. Wood (2000). Architectures to make Simple Visualisations using Simple Systems. Proceedings of Advanced Visual Interfaces - AVI2000, ACM Press, pp. 51-60.

[[D02]] A. Dix (2002). Managing the Ecology of Interaction. Proceedings of Tamodia 2002 - First International Workshop on Task Models and User Interface Design, Bucharest, Romania, 18-19 July 2002

[[GBK99]] H-W. Gellersen, M. Beigl, H. Krull (1999). The MediaCup: Awareness Technology embedded in an Everyday Object. Int. Sym. Handheld and Ubiquitous Computing (HUC99) Karlsruhe, Germany, 1999

[[H90]] 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.

[[NEL91]] Newman, W., Eldridge, M. & Lamming, M. (1991). Pepys: Generating Autobiographies by Automatic Tracking. In Proceedings of the second European conference on computer supported cooperative work – ECSCW '91, 25-27 September 1991, Kluwer.Academic Publishers, Amsterdam. pp 175-188.

[[N90]] D. Norman (1990). The Design of Everyday Things. Doubleday, New York.

[[S87]] Suchman, L. (1987). Plans and Situated Actions: The problem of human–machine communication. Cambridge University Press,.

[[WHFG92]] Roy Want, Andy Hopper, Veronica Falcao, Jonathon Gibbons. The Active Badge Location System. ACM Transactions on Information Systems, Vol. 10, No. 1, January 1992, pp 91-102.