Chapter 1

Theoretical Analysis and Theory Creation

Alan Dix (2nd August 2007)

1.1 Overview

"*the initial impetus for research is the search for theory*"
(Fawcett and Downs, 1986)

A chapter on theory as a research technique is strange as, in a way, what is academic research about if it is not about theory – without theory we may be engaged in product development, or data gathering, but not research. This said, there is of course also a *spiritus mundi* against theory: in abstracting away from the particular, theory is seen as at best simplistic and at worst reductionist and dangerous. And of course in popular language a theory is an unsubstantiated guess, almost the opposite of the scientific understanding of theory!

A theoretical approach is also not so much a method or technique that is applied to research, but an attitude and a desire to make sense of and to understand, in some ordered way, the phenomena around us. This approach can influence design and research methodology; indeed those most avowedly atheoretical in their methods are often most theoretical in their methodology!

Theories, that is systematic and structured bodies of knowledge, are the raw material for both research and practical design, but are also the outcomes of research and often the results of more informal reflection on experience. As we shall discuss shortly, theory is the language of generalisation, the way we move from one particular to another with confidence.

And theories are more basic still. A tiny baby watches her moving fingers, hits out at a ball and sees it move, gradually making sense of the relation between feelings and effects; the building, testing, and use of theory are as essential a part of our lives as feeding and breathing.

In HCI, developing, understanding and applying theory is particularly important. Technology and its use move so rapidly that today's empirical results are outdated tomorrow. To be proactive rather than merely reactive and to produce research results that are useful beyond the end of the current project or PhD requires deeper knowledge and informed analysis.

In this chapter we will first spend some time examining what theory is about: why it is important, what it is and is not, and at different kinds of theory. We will then look at different ways of using theory in HCI practice and research and ways of producing new theories. These techniques will be demonstrated by a number of real examples in research and commercial practice. Finally we will look at some of the strengths and weaknesses of theoretical approaches and the way they relate to other techniques.

1.2 About Theory

1.1.1 Why theory?

As we have noted, theory is the language of generalisation. There are two main ways in which we can generalise from past experience in order to deal with new situations:

analogy – here one looks for a past situation that is similar to the current one and then tries to use parallels between features of the past and current situations in order to explain aspects or predict outcomes.

abstraction – here one tries to find common features from past experience and use these to create more abstract concepts. When new situations arise they are expressed in terms of the abstract concepts.

We do both these things in day-to-day life and are usually not aware that we are doing them. In particular, our brains do a lot of analogical 'reasoning' in the form of associative pattern matching at subconscious levels including interpreting raw perceptions. In any analogical reasoning there is already an implicit theoretical model that determines what is considered important in comparing two instances. However, it is when we give names to the attributes and classes that we move from analogical to theoretical thinking.

If we say "drunk drivers are dangerous because they take risks and have slower responses", then we have taken experience (our own, national statistics, medical facts) and put it into concepts ("drunk driver", "risk") with relations between them – a theory. This language of generalisation makes it possible to reason more explicitly about the subject of theory, as well as communicate general observations. Theory is not just the language of generalisation; it is the language of technical culture.

The main difference between work-a-day theories (like the above) and academic theories is that in the latter we take an additional step of identifying and naming the collection of concepts and relations that make the theory. When the theory is given a named identity it can then become the subject of more explicit critique and analysis. It is discussable, refutable and sometimes even provable.

Theory and analogy interweave. We often derive our theories after previous more analogical thinking, so analogy helps abstraction. But also theory helps analogy: if we have ideas of abstract concepts and properties, we can use these to explicitly retrieve previous examples ("I recall a case of a drunk driver last year …") and then use the retrieved examples to draw conclusions that perhaps the theory does not (yet) support. Furthermore, having seen a past example that is similar to a new one, we need to adapt the past case, and to do that, we need to understand sufficiently the implications of the differences – which is precisely what theoretical understanding helps us to do.

The power of abstraction is not just about communication, but also that it gives us the means to reason about things. As we shall see later this has enormous practical benefits in HCI research and practice.

1.1.2 What is theory?

Theory is a hard word to define. As noted, in common language a 'theory' is a hunch or bright idea. But scientific theories are very different. The *Oxford English Dictionary* has a number of definitions of different uses of the term (including the common use one!), but the one most pertinent to scientific theory is:

> *"A scheme or system of ideas or statements held as an explanation or account of a group of facts or phenomena; a hypothesis that has been confirmed or established by observations or experiment, and is propounded or accepted as accounting for the known facts: a statement of what are held to be the general laws, principles, or causes of something known or observed."* (OED, 1973)

If we were describing a theory in the social sciences, we might use slightly different words, but the essentials are the same:

structure – A theory is not just an isolated set of statements, laws or facts, but has some level of structure or inter-relation.

explanation – Because of this, theories can be used to give an explanation of *how* or *why* things are true, not just *what* is true.

abstraction – Theories account for more than single observations, but offer more general or abstract accounts.

verity – However, theory is not divorced from particulars; it is usually based on experience or observation (induction) and can be used to explain or predict future observations.

The last of these is the critical difference from the common use of the word. While theories differ in the extent to which they have been justified or verified, they must all have some relation to reality, not mere hunches. However, theories may be wrong. For example, I might have a theory that gravity is caused by the pressure of air holding you down and that as you go higher up the air pressure reduces and hence so does gravity. This happens to be a false explanation of gravity, but is not without some justification from experience (air pressure is indeed powerful, although caused by gravity rather than the other way round). Critically, the theory has enough explanatory power to make predictions, for example, if we make a vacuum where there is zero air pressure, then things will be weightless in it. Because we can make predictions we can then test this theory, and in this case find it was wrong.

Karl Popper, a philosopher of science, regarded this ability to test scientific theory, *falsification*, as being a key element that distinguishes true science (Popper, 1959). By this criterion, if it is inconceivable that a theory could be disproved, then it is not a scientific theory. While this captures some aspects of science, in fact science often works by holding onto theories and modifying them rather discarding them 'just' because the evidence does not fit. This is partly a human tendency towards conservatism and partly a sensible use of the intellectual investment of a community.

Theories may be built upon other theories or laws; for example, the Model Human Processor (Card et al., 1983) is a theoretical model of human information processing built upon older psychological theories and results. I might then construct a theoretical account of text entry on mobile devices based on MHP and other knowledge of mobile use.

1.1.3 Things like theories

Theories differ from *laws*, which are also abstractions over observed phenomena, but do not offer systematic explanations for the phenomena. For example, Boyle's Law says that if you double the pressure on a gas and maintain the temperature it halves in volume, but does not say *why* this is true. Similarly Fitts' Law says that the time to hit a target varies logarithmically with the distance to the target, but again, as a formula, does not say *why* this is true.

Theories do relate to laws. Theories may explain laws: for example a molecular theory of gases where the gas molecules cause pressure by bouncing off the walls of a container can be used to explain and derive Boyle's law. Similarly, you can offer theories of Fitts' Law in terms of information theory (as in Fitts' original paper (Fitts, 1954)) or in terms of the feedback loop between perception and action (Keele, 1968; Meyer et al., 1988). Theories may also build on laws: for example, we may use a combination of Boyle's laws and observed data on the calorific value of coal to develop a theory of steam engines. Similarly, we may use Fitts' Law as part of a theory of pen-based mobile interaction.

Theories also differ from *paradigms*, which are more 'world views', ways of looking at things. Theories are framed within a paradigm, as it is the paradigm that gives the theory a way of looking at phenomena, and so for someone outside that paradigm a theory may be hard to comprehend (imagine explaining quantum mechanics in the 14th century). There may be many theories operating within a single paradigm, and theories may develop and change while still operating within the same paradigm. We have already noted that real science tends to modify rather than discard theories, but sometimes as an area develops it gets harder and harder to 'fix' the theories until eventually a whole new way of thinking about things is required. Thomas Kuhn, another philosopher of science, identified this phenomenon; he describes these times of critical change as 'scientific revolutions' with associated 'paradigm shifts' in the way people think (Kuhn, 1962).

Perhaps the most well-known example of paradigm shift is the change from a Newtonian worldview in the early years of the 20th century. Until this point, the very small and very large are seen as operating very much like the 'normal' world we experience (a form of reasoning by *analogy*). However experiments were increasingly at variance with the associated theoretical models (problems of *verity*) until the point at which the Newtonian worldview was challenged and superseded by Einstein's General Relativity (for the large) and Quantum Mechanics (for the small).

In HCI a similar (although less dramatic) shift happened in the late 1980s when the more cognitively based theories of early HCI were challenged by more situated accounts in Winograd and Flores' *'Understanding Cognition*' (Winograd and Flores,

1985) and Suchman's '*Plans and Situated Actions*' (Suchman, 1987). Because theories in HCI are by their nature partial and approximate, the disparity with the 'facts' of observation is less clear and so, rather than replacing an 'older' view, this is more an additional way of looking at things. Naively, one might imagine that when the 'truth' of theories and paradigms is partial, or not easily verified against the world, academics would tend to use them more pragmatically. However, the opposite tends to be the case and in HCI (as in the social sciences) one tends to get partisan adherents of one viewpoint or other!

Finally, another related idea is that of a *model*. Minsky (1965) defined a model using the following "*To an observer B, an object A* is a model of an object A to the extent that B can use A* to answer questions that interest him about A.*" This definition is remarkably similar to attempts to define metaphor or analogy, and indeed, writing from a social science perspective, Hawes defines a model precisely as an analogue. Models are similar to theories and operate at a similar 'level'; however, whilst models, like laws, are mainly about telling you *what* is true, theories are more about *why* things are the way they are; or as Hawes puts it "the former [theory] is an *explanation* whereas the latter [model] is a *representation*". A model is often an embodiment of a theory. For example, the explanation of Boyle's law in terms of a molecular theory involves a model of molecules hitting the walls. The *theory* is the combination of the idea of having macroscopic phenomena explicable through microscopic causes, the use of Newton's laws for the individual molecular collisions, and rules about the relationship between temperature and velocity of molecules. This theory is then embodied in a mathematical model, which can be used to derive the laws.

The boundaries between these terms and others such as 'frameworks' is somewhat fluid, and it is not uncommon to find referees of papers arguing that what an author has called a model is actually a framework, or an architecture, or something else. Whilst it is important to be careful in your terminology, it is more important to know whether you have understanding that is sufficiently deep to be able to explain the phenomena you see and perhaps predict what will happen in new, unseen and potentially radically different situations.

1.1.4 Types of theory

There are many types of theory, from the mathematical equations of general relativity to theories of social relationships.

One important distinction is in the *generative power* of the theory:

descriptive – given a cause and its effect, tells you why it happened. This kind of theory is applied after you have observed a phenomenon and allows you to explain and make sense of what you have observed. The danger of purely descriptive knowledge is that it can often be 'twisted' to explain any results (failing Popper's falsification test), so you have to be very careful in formulating and applying it so as to avoid this.

predictive – given a cause, tells you what effect will follow. This kind of theory can be applied before you have seen the effect of an action. It is the point at which most sciences stop. With a predictive theory of some aspect of user interaction, you are able to look at a design and say "ah yes, this will/is likely to happen". Evaluation techniques such as cognitive walkthrough or heuristic evaluation, whilst lacking the *structure* of a theory do have this form of predictive power.

synthetic – given a desired effect, tells you what to do to cause it. This last form of theory is most useful in design and engineering … but least common. Like predictive theories, it can be applied before you have observed a phenomenon. However, in addition it can be used *backwards* to ask, "I want this to happen, what should I do to make it happen?" In user interface terms this may be "I would like users to enjoy/be efficient with this interface, what should I design to achieve this". It is in this highest form of knowledge that the power and importance of theoretical understanding is most clear.

Where the space of potential causes/actions/designs is small it is often possible to move from description to prediction and from prediction to synthesis through "what if" thinking. However, once the complexity of the design space becomes large this becomes impossible.

We can also characterise theories by the kinds of knowledge they provide along a number of dimensions / categories:

qualitative vs. quantitative – Some theories deal with precise numbers (time, error rate) or clear countable categories (Male/Female), others with more qualitative concepts such as happiness.

precision – Actually there are a number of different criteria that lie under this heading. Are the predictions/explanations precise or approximate, deterministic or probabilistic. In a more qualitative theory does the theory say 'sometimes' or 'always'? Theories do not need to be precise to be useful: when buying a car it is useful to know that a Porsche is more likely to win a race than a Lada, even though, occasionally, through breakdowns, or better driving, the Lada may win.

aggregate or individual – We may be able to say things that are true of each user, "interface A is always faster to use than interface B", but some theories may only tell us about aggregate or average properties: "most users will find interface A faster than interface B". Even for a single user, some things may hold true for every interaction, whilst others only apply to averages. For example, a user may generally prefer one device to another, but under certain (uncommon) lighting conditions the opposite may be the case.

scope – Is the theory universal, applying to everything, or are there limits.

While scientific theories are often associated with predictive, quantitative, deterministic, universal knowledge, in fact many combinations are possible:

- *The probability of a die falling on '3' is exactly 1 in 6* – universal, predictive, quantitative, probabilistic knowledge.
- *There is no smoke without fire* – universal, predictive, qualitative, deterministic

- *Small targets on average take longer to select, so to speed up data entry use larger buttons* – aggregate, synthetic, quantitative data (size and time measurable), but not precise (just longer' not how much longer)

These are often linked. In particular it is common in HCI to have qualitative theoretical explanations of quantitative measurements.

1.3 The Method

Theories are so varied it is hard to talk about a single 'method'. We will look at three facets: the relation between theory and empirical methods, methods for deriving theories, and finally the relation between abstracted theoretical understanding and specific situations. This is not exhaustive, but gives a start point for applying and delivering theory.

1.3.1 Theory and empiricism

In the physical sciences measurements are taken so that they are as ‘pure’ as possible in order to reduce the possible causes of an observation to the single phenomenon being studied. Psychological experimentation emulates this, although the complexity of the human subject makes assigning single causes more difficult, but given this constraint the aim is again to minimise the possible causes. In HCI we may wish to experiment in near-real settings as we know that the effects seen in the laboratory are often very different from those in the field. Even when we experiment in a laboratory we will still use realistic tasks and user interfaces. This leaves us with multiple causes: the task chosen, the context of the experiment, the fine details of the prototype interface, the chosen user group … and amongst all this the actual effect we wish to study.

Theoretical understanding can help to unravel this knot of potential effects. Most important is understanding *mechanism* – the details of what goes on, whether in terms of user actions, perception, cognition, or social interactions.

Theory can help in the design, analysis and application of experiments and empirical data gathering:

design of empirical studies – If we understand the details of how we expect a user to interact we can predict what tasks will amplify the desired effects and minimise the 'noise'. For example, as soon as people need to *think* in a task, the variability between people (and between runs with the same person) increases dramatically, which in turn makes it difficult to run experiments that yield statistically significant results. However, careful choice of tasks and measures can increase the likelihood that effects can be measured. In qualitative experiments also, understanding what is likely to happen enables you to choose appropriate measurement techniques (video, keystroke logging) and tasks that again make it easier to see effects.

analysis of empirical data – Experiments are often reported in terms of end-to-end measures such as overall error rates, or task-completion times. Because there are often interaction effects[1] between the many phenomena affecting the data, these end-to-end measures are often as much about the particular choice of task or system as the target phenomenon. If you understand the details of the mechanism, the steps and phases of the interaction, you can choose finer measurements and use these.

application of empirical data – From experimental data alone it is possible to interpolate between measured values. If a user takes 30 seconds to do 2 tasks and 50 seconds to do 4 tasks, it is reasonable to expect 3 tasks to take about 40 seconds or at least something greater than 30 and less than 50. Even here you need to know that there are not likely to be any odd intermediate effects. However, to extrapolate is far more dangerous: no tasks would probably take no time, not 10 seconds! There are often limits to effects where things get much harder or easier. However, with some theoretical understanding of the underlying mechanism we can know whether to expect such limits, and whether extrapolation beyond the bounds of our empirical data is likely to be sound.

recontextualisation of empirical data – As well as extrapolation beyond the quantitative limits of our data, we typically want to extrapolate from a study performed on a particular interface with particular tasks, to similar interfaces and similar tasks. Again it is the theoretical understanding of *why* we see the effects that we see that enables us to make this generalisation.

synthesis of empirical data – In practical design situations we typically know bits of psychological knowledge, data or models of the system behaviour, perhaps previous experimental evidence related to parts or aspects of the system we envisage. If you have some idea of how things work it is easier to bring these together.

To see several of these aspects at play, consider a recent experiment where we varied delays during web page navigation. In addition to end-to-end timing measures, we also measured the average time users took making decisions at intermediate 'menu' pages (time from page presentation to menu selection). This decision time increased markedly when the web site was slower, showing that people adjust their behaviour. The end-to-end timings were much less clear, as they were influenced by effects such as the number of pages that the user visited.

The choice of the decision time as a measure was not arbitrary. A previous experiment had observed improved learning of menus when delays were longer. Constructive learning theory suggests that the more you 'work' on information, the

---

[1] Here this means interaction effects in the statistical sense (see Chapter 99). An interaction effect is where two conditions cannot be treated entirely independently. For example, in an study we might find that women are wiser than men and that people get wiser as they grow older – independent effects. Together these might go some way to describing the data; however, it may be that even when we take both these effects into account women get wiser more rapidly than men – this would be an interaction effect.

more integrated it becomes into your personal knowledge structures and hence you get better learning. We hypothesised (based on cognitive ideas of effort minimisation) that users would think more carefully about decisions for the slower interface. This is because the 'cost' of failure (waiting for the page and then hitting 'back') would be greater when the delay was longer, therefore the menu of the slower interface would be more thoroughly processed and hence more effectively learnt.

So, here we see theoretical understanding being used to synthesise existing theoretical and empirical results in order to analyse the results of the first experiment, and this then being used in the planning of a further experiment and the effective choice and interpretation of measures.

1.3.2 Developing theory

Theory constriction is a creative process, so there are no simple handle-turning rules to create a theory. However, there are a number of techniques that can help you.

abstraction and organising – Those tables and taxonomies that appear in so many papers are not just ways of laying out information to make it more readable. In addition, they create terms, concepts and categories that form the basic vocabulary of many theoretical descriptions. These taxonomies, dimensions, etc., may be drawn from professional experience, existing theoretical knowledge, or different forms of primary data. The concepts on their own usually generalise over many instances, and if the concepts are gathered into some form of classification or taxonomy, then they provide the means for further generalisation. However, while taxonomies are useful for giving an overview, the greatest analytic power comes when there are multiple simultaneous classification schemes (dimensions, simple categories or taxonomies). With multiple classifications, you can tabulate one against the other and look at how examples (transcript utterances, previous systems, experimental data) fit into the table. Where there are gaps this may signify impossible situations, or potential for novel solutions. Where there are patterns this may suggest systematic relationships (see Fig. 1).

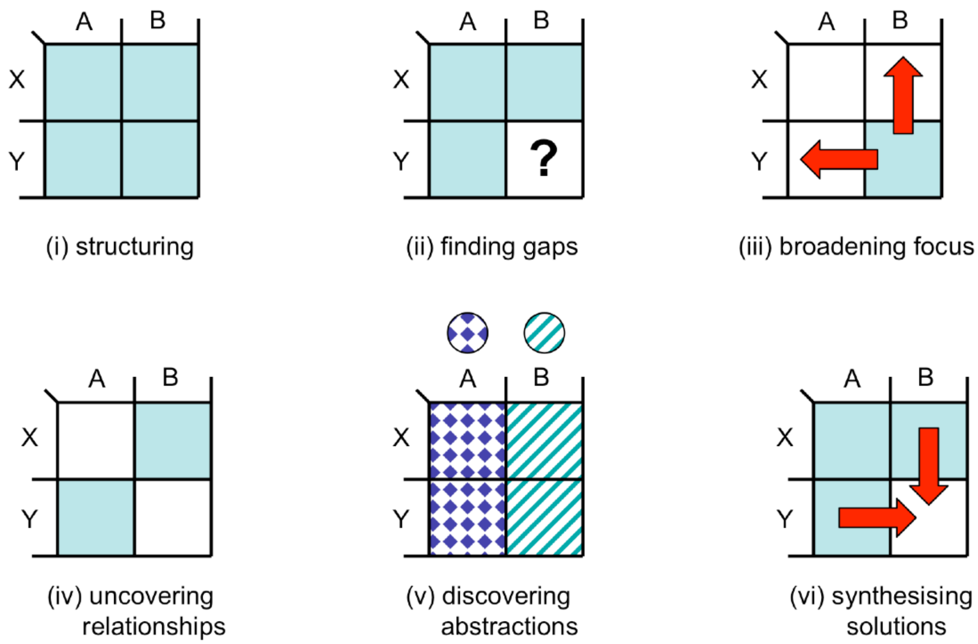

Figure 1. Uses of multiple classification (from Dix, 2002)

exploring definitions and boundaries – Academics also love precise definitions and they are of course important to help make sure we share a common vocabulary. Definitions also enable one to draw precise bounds on concepts, but in a human-centred discipline like HCI, these hard-edged categories often stand at odds with the more nuanced realities of situations. This can be seen as a 'problem': either requiring more rigorous and careful definition, or as an argument or reject the process of definition *per se*. However, it can instead be seen as an opportunity. The boundaries at the edge and between categories are often the most fruitful areas to learn about them. Look at the terms, dimensions and properties you have used to formulate the definition. The definition itself is not the real value – this vocabulary is. By attempting to articulate the criteria that delineate the edges of the category you have learnt about the properties that characterise and explicate its heart! The power lies not in definition (which will be wrong!), but in the *activity* of definition.

critical transitions – This is a particular form of boundary exploration that can be used to explore categories that you can recognise when you see them, but find hard to define more precisely (we have used this for things like fun!). You choose one example that is in the category (e.g. playing party games) and one that is not (sitting an exam) and then produce a series of intermediate examples between the two (sitting an exam in a party hat, playing *Trivial Pursuit*). As you trace the trajectory there comes a point when you start to say that the examples cross the boundary (between fun and non-fun) – the critical transition – and then you can ask yourself "what happened", and so uncover critical criteria, dimensions or properties of the core category.

child-like questions – Listen to a small child and they constantly ask "why? why? why?"; this is the intellectual equivalent of the baby playing with its hands.

Each question probes and builds richer models of the world. And the child is never satisfied with the first answer, always probing deeper. Why is it dark at night? Because the sun goes down. Why does the sun go down? Because the earth turns round. Why does the earth turn round? We can emulate this and, like the child, not be satisfied with first answers. Why is this interface better? Because it is consistent. Why does being consistent make it better? Because … you answer that one! Children also ask "What if?". What if we were on Mars, would it be dark at night? What if the earth didn’t turn round? Again we can ask similar questions. What if we used this for a different purpose? What if the user gets interrupted? Rather like exploring boundaries, it is not so much the answers that are important as the understanding we gain in the process. In particular, this form of thinking often helps in establishing causal and other relationships between core concepts.

formal representations – Some theoretical understanding can be embodied in some form of formal or mathematical model. For example, if we are looking at selection from menus, we may model this as a sequence of actions of the form (i) visually scan menu for right item, (ii) select option with mouse, (iii) wait for screen to refresh, (iv) repeat for next menu level. Given this model, stage (i) will be linear in the number of menu items, stage (ii) is a Fitts' Law task and (iii) is measurable for a given system. So we can build a model of how long selection will take from a particular menu hierarchy, and hence give ourselves a way of comparing different choices. Even more qualitative theories may benefit from more structured representation. For example, rather than a precise set of equations we can follow a diagram such as Figure 2 to see that increasing the number of items on screen will increase the time taken to make a selection.

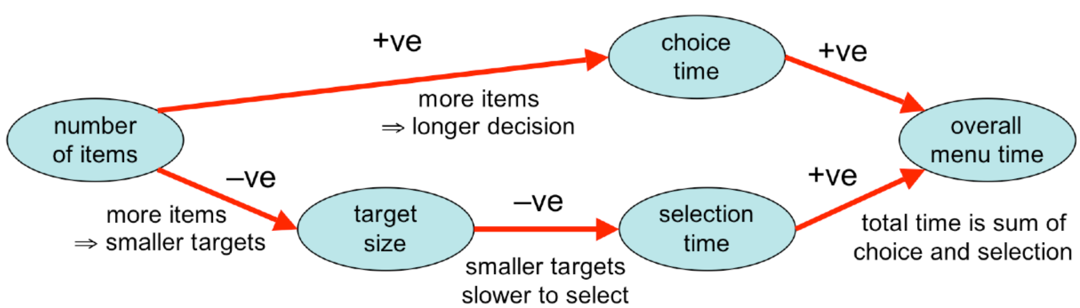

Figure 2. Network of influences of number of items shown on screen

recoding dialectic – It is often hard to verify qualitative theories, especially to know if they are complete – have you entirely missed some important issue? Even when theories are built inductively from large data corpora, it is hard to know whether you have imposed some blinkers or pre-existing ideas that meant you missed things out. Recoding dialectic is a way to help give yourself confidence in the completeness of a qualitative theory, and to help you fill in the gaps you discover. All you do is go back to the primary data, whatever it is, perhaps utterances in a transcript, or results of a literature review, and, for each item, describe it using the vocabulary of the theory. Imagine a user has said "I hate this system because it keeps logging me out". You might encode this as "timing problem in authentication procedures". Sometimes you find that you cannot describe the utterance using the vocabulary. This might be because the utterance is irrelevant, but if not it suggests a gap in the theory. If you are able to describe an item using the vocabulary, you then say to yourself, "this just says …" – and feel the tension the word 'just' gives you. Is that really all the statement says? If the description using the terms in the vocabulary feels inadequate, try to describe *why* it is inadequate and in that description you often find the seeds of new concepts or relationships that belong in the theory.

As noted, these techniques are not a guaranteed way of building theories, but they are heuristics that can help. Look also at papers and books that describe new theories, or theoretical concepts; often these are presented as a *fait accompli*, but sometimes they also describe the process by which the insight was found.

1.3.3 Theory and the particular

We discussed earlier the paradigm shift in HCI in the later part of the 1980s with a move from more mechanistic models of human cognition to more situated and contextual models. For many this has led to a distrust in generalities and theoretical descriptions – each situation is different and special; any form of generalisation will miss important details.

Certain forms of ethnography (in particular the ethnomethodological school (Heritage, 1984), which has been so influential in HCI) can be particularly antagonistic to theory, largely in reaction to the perception that social science theories are often foisted onto the real world rather than expressing it. However, this stance is perhaps a little disingenuous, as there are clear theoretical stances underlying ethnomethodologically informed ethnography, in particular the focus on social accountability. In addition, the rich accounts and vignettes, while expressing particulars, are clearly also chosen to highlight behaviours and issues that one expects to see repeated elsewhere (generalisation through *analogy*).

Three strengths of ethnographic approaches are:

- *starting with real data* – While no human observation is without preconception, the aim is to be as open as possible to what is observed.
- *limited generalisation* – While ethnographers bring with them past experience of similar contexts and situations, they expect any similarities to be limited and partial.
- *reflexive practice* – Good ethnographers are aware that they do bring preconceptions and that they do influence the situation and so explicitly take this into account in their analysis.

Whilst eschewing theoretical approaches entirely seems an unproductive over-reaction to its dangers, certainly these strengths are ones that are worth bearing in mind in more theoretical approaches. In particular, it is easy to forget the limited nature of more formalised theories.

One technique from the social sciences that is gaining support in HCI is *grounded theory* (Glaser and Strauss, 1967). Like ethnographic techniques it is focused on taking a very open approach to raw observation and experience, but uses this in order to construct theoretical descriptions from the particulars of the observations. Its techniques are very much those of *abstraction and organising* described in the last section. In its purest form, it starts using open coding of the raw data: writing whatever terms make sense to describe each utterance or data item. These terms are then gathered, sorted, organised into taxonomies and relationships until a more abstracted theoretical account is created. While the standard methods for this do not include a specific validation step, *recoding dialectic* is directly applicable.

As well as building more general theories from observations, you may simply need sufficient theoretical understanding for the problem at hand – a *situated theory*. This is not an attempt to build a new theoretical framework that will generalise to different situations, but instead a building of theory (usually from existing pieces) that is just for what you are doing now. It may then turn out that this theoretical account generalises, but that is not its primary purpose.

Some while ago I was talking with a cleaner in our office block and we were discussing the problem of the paper getting unravelled in the toilet roll holders. The holder contains two toilet rolls, one above the other, so that as the paper is pulled from the lower roll, the upper one turns with it. We realised that a toilet roll unrolls if turned in one direction, and is held tight if turned in the other. If the toilet rolls are inserted in the same direction then the upper one will unroll with the lower one and get clogged up with loose paper; however, if they are in opposite directions then the upper one will be held tight as the lower one is pulled out. We had created a *situated theory* of toilet roll unravelling. In fact the theory was not ungeneralisable and the cleaner drew an analogy with the nuts on bicycle wheels, which are designed to tighten rather than loosen as the wheel turns, but this generalisation was not essential to solve the problem at hand.

Perhaps the hardest aspect of performing theoretical analysis is to hold *both* the abstract and the particular in mind at once. It is easier to either look only at the particular, or only at the abstract, but the greatest gains are usually from keeping both in mind. Even in school geometry you always draw a diagram – a particular example – and this helps you to think about the abstract proof.

1.4 Applying the Method

Much of this discussion has been quite abstract, but now we will look at a few examples of how theory can be applied in the practice of research and design. We will look at three different kinds of application:

- applying theory to the design of a new product
- applying theory to understand design methods,
- applying theoretical thinking during analysis in order to develop new theories

These do not cover all the possible ways in which theory is used, but will give some indication of how theory can be used in practice.

1.4.1 Applying theory to design

From 1998–2000 I was involved in a dot.com company, aQtive, and our main product was an 'intelligent' internet desktop agent called onCue. The underlying software architecture was heavily influenced by status–event analysis, which we will discuss later in this section. However, for now we will focus on the user interface itself.

onCue usually sat as a small toolbar on the side of the screen, and whenever the user copied any text to the clipboard, onCue examined the text and then changed the toolbar to reflect the contents. If the text was a postcode onCue might suggest an online mapping service; if it was a table of numbers, onCue might suggest adding them up or making a graph.

Of course the intelligent recognition algorithms behind onCue were not perfect, and we knew there was a history of failures in intelligent interfaces. In trying to make sense of how to design onCue we created a *situated theory*. In broad terms we realised that the important thing was to define the interaction around onCue so that when the recognition was wrong (which inevitably it would sometimes be), the interface would not be annoying. This we later formulated into the three rules of *appropriate intelligence* (Dix et al., 2000), the generalisation of our situated theory, listed in Figure 3. The first two are what you normally expect of an intelligent system, and look good in demos, but it is the last that makes a system work in practice.

1. be right as often as possible
2. do something useful when you are right
3. when things go wrong, don’t mess up the user

Figure 3. The three rules of Appropriate Intelligence (Dix et al., 2000)

As well as being 'intelligent', onCue was also proactive, and might make suggestions even when the user did not want them … so rule 3 was especially important. We needed to design onCue to be always readily available, but not to interrupt or distract the user – no bouncing Clippy.

First we made it non-modal (already a UI theoretic term) and small enough that it could be 'always on top', but not in the way; the user could choose to shrink it to a single small icon. When the user cut or copied to the clipboard, the contents of the onCue toolbar changed. Even though onCue was not modal, still there was the possibility of visual distraction. However, we knew that the fovea is very narrow and so the toolbar at the edge of the screen would be in peripheral vision. We also knew that in peripheral vision rods predominate and these are most sensitive to rapid movement and change. So instead of simply switching the icons in the onCue toolbar as the clipboard changed we slowly faded out the old icons and then faded in the new ones over a period of about 1 second. While this was fast enough for there to be no real delay for the user, it was slow enough that it did not register as a change in peripheral vision and so was not distracting.

Note how we made use of theoretical understanding of the human visual system as well as the situated theory of appropriate intelligence in order to make an effective design.

1.4.2 Applying theory to method

Personae and scenarios are heavily used in HCI practice. However, these vary substantially in detail. Some people create personae that are not very different from a user profile, "Mary is a 30 year old office worker with 2 children and a dog", while others create rich descriptions like a character in a novel. Similarly, scenarios vary from a bare list to something more like a short story. What is the right level? Is there any value to the irrelevant details that litter the richer descriptions?

Although the details are debated, it is generally accepted that in addition to logical/rational thinking, we also have different kinds of specialised knowledge at work, so-called 'multiple intelligences' (Gardner, 1983). For example, for interpersonal relations, we recognise the social intelligence that enables us to know what to do, or related emotional intelligence that gives us insights into what others feel; in contrast, when doing more physical things, we have an understanding of our own bodies and also of natural objects. These ‘intelligences’ work together in day-to-day life and in particular may allow us to make intuitive or subconscious judgments as well as more explicit reasoning.

When we deal with very sparse propositional descriptions of the world, only the explicit encodings of these specialised forms of knowledge are available: "person meets friend implies person smiles". Yet we respond in real life more naturally and immediately than this. However, if we create a rich imaginary picture of a scene – imagine yourself meeting a long-lost friend – then these intelligences naturally swing into action. When you read a novel you may think "Daniel wouldn’t say that"; you 'know' the character and so know what he is or is not likely to do. If a person is acting 'out of character' in a novel, either (like real life) something is wrong with the person, or the author is not very good.

It is precisely the irrelevant details in a rich persona that allow us to 'get to know' the person and therefore be able to respond to the character in a scenario and say "Mary would press that button now" or "Mary wouldn’t understand that". Similarly the images elicited by a rich scenario allow your physical and spatial intelligence to kick in: "he couldn’t do that; it is too far across the room".

See how theories of multiple intelligence allow us to understand *why* rich personae and scenarios work, and thus why it is important to retain 'irrelevant' details. Furthermore, because of this understanding we can consider other ways of recruiting a designer’s multiple intelligences; for example, I suggest that students physically act out scenarios.

1.4.3 Applying theory during analysis (making theory)

Some years ago I was involved in formal modelling of user interfaces. Models of keystrokes worked quite well: each key caused a change of state of the system and a resulting change in the display. However, things were not so good when it came to the mouse. Although you could model the mouse, a series of tiny movements, each making a small change to the mouse pointer on screen, it did not 'feel' right; when you move a mouse it does not feel like a lot of tiny movements, but one continuous gesture.

This insight led to *status–event analysis*, the recognition that some phenomena, such as the screen contents and the mouse position, always have a 'value' – they are *status*, whereas others, like the key-presses, or beeps from the computer, happen at a particular moment – they are *events*. This is actually very obvious, and yet different from the event-dominated descriptions that are common across computing. This insight was first used in the formal modelling of mouse-based systems and later became a generic way of looking at a wide variety of phenomena.

One of the powerful things about status–event analysis is that it applies at many levels, from the internal electronics of the keyboard to social interactions. Actually, you should be cautious when you hear a claim like this; theories of everything usually turn out to be theories of nothing. However, this time it really did work (honest!). Using status–event analysis, not only can one examine phenomena at these very different levels, but often one can see similar behaviours at the different levels and with both electronic and human actors. For example, imagine you have to catch a train. You keep an eye on the time so that you leave on time, but you do not continuously look at the clock, every so often you look to see if it is the critical time. This is a polling behaviour, and the same technique is used in network applications to look for changes in remote databases, and by a keyboard controller when it looks to see if one of the keys is pressed.

Notice how a formal modelling of the interface led to a situated theory to deal with a particular problem. The problem came to light because the more abstract formal representation was considered alongside the 'gut' reaction to the real situation. When the disparity was noted it did not lead to an abandonment of the formal theory, but instead to an attempt to understand the nature of the discrepancy (*boundaries and edges*). This in turn led to both an improved formal model and also the formulation of a general qualitative theoretical position.

1.5 Critique

1.5.1 Focus or blinkers

We have already discussed one of the main criticisms of theoretical approaches, raised particularly by the ethnographers: by abstracting, theoretical views act as blinkers, meaning we only see the things that fit the theories and miss the rich details of each particular situation.

To some extent the examples in the last section show that it is possible to hold on to both the particular and the abstract and gain insight by doing so. However, it is a real danger and many people tend to separate the two. Academic disciplines in general and computing in particular tend to suggest that more abstracted thinking is good thinking. Of course this is true academically, as we need to abstract in order to generalise, but in addition we need to keep those abstractions continuously grounded.

If you do not naturally keep both in mind at once then you need to explicitly alternate between the two, sometimes reflecting on particulars and attempting to see the more abstract picture, and sometimes looking from your abstractions and seeking concrete examples (although this can be hard, see Dix (2007)). In particular, you need to be constantly on guard against trying to fit the world to your preconceptions, and instead actively look for discrepancies, not as 'faults' in your theories, but as opportunities to learn more.

1.5.2 Is it practical?

Theoretical is often seen as the opposite of practical. Again, the examples in the previous section show that this need not be the case. There is the old saying that "those who can do, and those who can't teach". Most academics, being teachers, would want to disagree with this, and in fact the knowledge required to teach something is often not the same as the knowledge or skills needed to do it. If we take a similar view of professionals and academics we could say:

professionals – do things

academics – know about doing things

The theoretical reflection embodied in academic practice can be an ivory tower discussion, but can also lead back into practice. Consider an Olympic runner and her trainer. The trainer may not be able to run himself, but he understands nutrition, and muscle groups, and has spent hours watching slow-motion footage of the runner. The trainer's theoretical knowledge *about* running helps the runner to run. Of course if the runner tried to think like a trainer while she was running she would probably fall over, but as part of reflective practice (how did that run go) and when things go wrong (why did I hit that first hurdle) more theoretical thinking comes into its own.

The onCue example shows this well. It is not necessary to think about the details of visual perception every time you choose an icon, but to solve a particular, difficult problem it can be very powerful to go back to basics. For more common decisions simple experience, standards or guidelines can help, but it is when you hit the unusual or apparently intractable that you need heavier intellectual armoury.

In HCI the most common development paradigm is an iterative cycle of prototyping and evaluation. However, iteration runs the danger of getting stuck in 'local minima' designs, which cannot be improved through small changes, and yet are not the best. This is particularly problematic when there are several facets that interact – for example, in onCue the slow change only works in combination with the positioning in peripheral vision and non-modal character. Theoretical understanding allows the designer to move from the *ant steps* of iterative design to *flea hops* – radically changing the design, based on deep understanding of what doesn't work and why (and therefore how to change it), and what does work and why (and therefore what should be retained).

1.5.3 Validating theory

Because theories are very abstract they can be difficult to validate. And if you don't know whether a theory is right, how can you trust it?

Whilst it is possible to validate theories, it is first worth noting that theories are rarely completely 'true'; they are views of the world and may be more or less complete, accurate and faithful and then only to some facet of some portion of reality. Nonetheless, they are useful. When you fly a plane the engineers will have used the theory of Newtonian dynamics that was proved 'wrong' at the beginning of the 20th Century, and yet, despite being known to be 'wrong' for 100 years, it is still useful, because it is *right enough* for the speeds and sizes of normal life. So when we apply theories, whether in research or practice, we always need to be aware not only of their fallibility and limitations, but also of their value.

We have already seen one theory validation technique, *recoding dialectic*. This was especially focused on the completeness of qualitative descriptions. When looking at a theory we can ask about:

correctness – does it accurately represent what it purports to

completeness – does it cover all the relevant phenomena and issues

It is typically the second that is hardest, since it is difficult to know what you don't know … as Rumsfeld put it, the "unknown unknowns".[2] This can be addressed both from the particular and from the abstract.

From the particular, one can test the theory against as many specific situations and examples as possible, looking for adequacy of explanation. It is best if these examples are generated from external sources, as of course you tend to think of those that are already covered by your own theories. For example, some years ago I was looking at the coping strategies people use when communicating over channels with delays that are badly matched to the nature of their collaboration. I outlined a number of strategies and then looked through the literature at snippets of computer-mediated communication reported by other authors. In each case I found examples of the coping strategies I had outlined, and did not find examples of other kinds of behaviour. This therefore increased my confidence in the completeness/coverage of the strategies.

Recoding dialectic, as discussed earlier, is a more systematic approach to checking the completeness of a theory against a known corpus. If, as is the case with

---

[2] Donald Rumsfeld was awarded the Plain English Campaign's "Foot in Mouth" award (Plain English Campaign, 2003) for his remark "… we know, there are known knowns; there are things we know we know. We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns - the ones we don't know we don't know." Although when spoken out loud it did sound rather enigmatic, and I do hate to defend Rumsfeld, I have never understood the hoo-hah about this. It is an important point and as clearly stated as I can imagine without far more words.

grounded theory, the corpus is the source for the theoretical constructs, then this is more of an *internal* check and should be supplemented by new cases. One technique used in quantitative methods (machine learning and statistical) is to deliberately hold back a portion of the data. You then analyse the rest to create a mathematical model, and test the effectiveness of the model on the 'unseen' cases. This can be applied equally well to experimental results, or even qualitative methods of analysis.

We'll turn now to *correctness*, which is perhaps easier than completeness, but nonetheless problematic.

In HCI it has almost become a joke that you cannot submit a paper without an 'evaluation' and that as long as referees can tick their mental 'evaluation' box the actual effectiveness of the evaluation is irrelevant. Now there is some truth in this as well as it being a caricature, and of course good referees look at papers on their own merits, not through preconceived templates. However, the truth behind the myth will continue unless as a discipline we understand what constitutes effective evaluation and, more importantly, validation of results.

The trouble is that many of the things we produce in HCI, including theories, are *generative artefacts*: things that are used (by people) to create other things. Theories are used to explain, to analyse, to design, to generate models, principles and even new theories. The success (truth, value, adequacy) of the theory is largely based on the effectiveness of the things it produces. Do the explanations it yields make sense? Do the things designed using it work? This is true of other things in HCI: guidelines, design principles, patterns, toolkits, algorithms and architectures. Even a particular system is generative in that the real thing of importance is the particular instance of its use.

For user interfaces we get round this effectively by sampling: user testing with sufficient users, ideally (although actually rarely) over sufficient tasks, in sufficiently varied situations. Unless you test every user, doing every task, in every specific situation (which by definition are unique) you cannot prove that the interface is effective, but with sufficient users you can make reasonable conclusions (and of course to know what constitutes 'reasonable' you take recourse to implicit or explicit theories!).

Where the artefacts that we are evaluating are further abstracted (e.g. theories, guidelines, toolkits), life becomes far harder. They are used by particular designers or interface engineers in particular circumstances to produce specific user interfaces. We can test the interfaces produced, but how can we know whether it is the theory (design principle, architecture, etc.) that was the cause, or whether it was the skill (or otherwise) of the designer? We would need to experiment with many designers being given many briefs (since briefs may be biased to particular techniques). The closest you see is when a class of students are used as surrogate designers and asked to use multiple techniques … although they are usually given a single brief and often know that one of the techniques is the 'pet' one of the experimenter.

In a recent paper about evaluation in visualisation research (Ellis and Dix, 2006), we characterised this problem as:

the evaluation of generative artefacts is methodologically unsound

It is not just hard, but impossible … and arguably *wrong*, as without great care in expressing your results such evaluations run the risk of appearing to produce general results that are in fact particular to the design situation tested.

Happily all is not lost; while it is unsound to believe you can empirically evaluate generative artefacts, you can *validate* them … and empirical evaluation can play a part in it!

Disciplines differ in the ways in which they validate their results; some focus on empirical testing, but not all – catch a pure mathematician doing statistics on a theorem … no, they *prove* them. You know the theorem is true, not because you try it out on lots of examples, but because the steps of the proof that led to it are its *justification*.

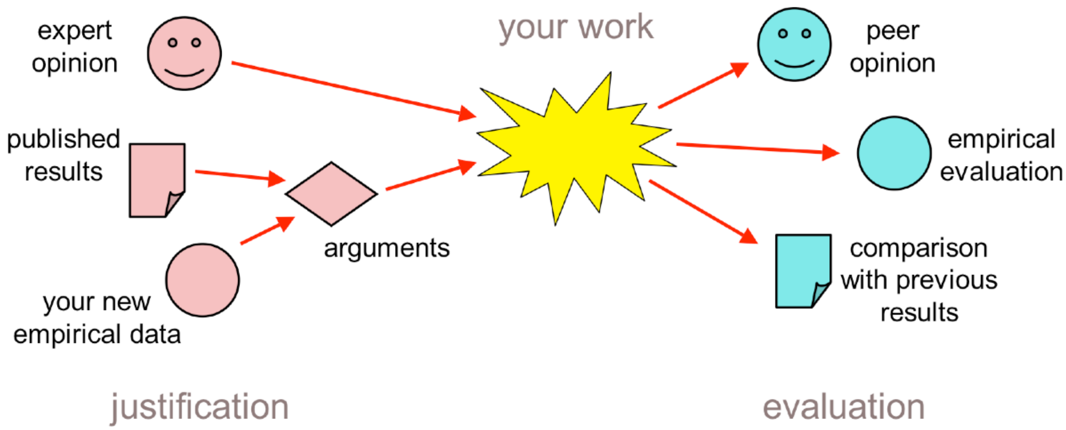

Figure 4. Validation from two sides

Because it is hard to validate generative artefacts, the justification becomes more important – why you thought the theory was a good idea in the first place. In mathematics the steps of the proof from axiom to theorem are indefeasible; if the proof is correct each step is entirely justified from its precursors. In most areas, and in particular HCI, this is not the case: our justification is sometimes from shaky premises (do we really believe the results of that experiment five years ago, which is poorly described?), or involves questionable deductions (can we really generalise from desktop use to mobile phones?).

While our justification will not have the decisive nature of a mathematical proof, we can make explicit and record the audit trail: the track of the argument from the assumptions we have worked on to the theory we have derived. The base points of this argument may be of many different kinds: previously published work by others, accepted practice, our own empirical data, or plain common sense (sadly undervalued!); and the 'reasoning' steps may range from formal deductions to a more vague feeling that something applies.

By knowing the strengths of different parts of the justification argument, we can know which aspects of our theoretical framework are strongly justified and which more problematic. Then empirical evaluation can be tuned towards the less well established aspects. In fact it is common to see the opposite, empirical work tuned towards the most established parts of a theory (or other generative artefact). This is perhaps good for writing papers (as we can formulate very strong hypotheses that are likely to be upheld), but it teaches us little (because the outcome is nearly certain). When empirical evaluation addresses the weak points of the justification, then the two work together to build our confidence.

1.6 Related Studies

Because HCI touches so many disciplines, there are many different theories that have been applied to it, from mathematical theories such as Game Theory or Graph Theory to social theories such as Actor–Network Theory or Structuration Theory. A good place to start is Carroll's '*HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*' (Carroll, 2003). This edited volume includes formal techniques, activity theory, information foraging theory and much more. Like this book, each chapter has a uniform structure, providing the scientific underpinnings of each theory or model, a detailed description of the theory, and one or more case studies. The case studies are particularly useful as they show how a wide range of different kinds of theory can be applied in practice.

In the early part of this chapter we looked at some of the broad philosophical discussions about the nature of science in the works of Popper and Kuhn. Within HCI there has been considerable debate about the nature of the discipline, most notably whether it should be regarded as science, craft or engineering (Long and Dowell, 1989). Unlike traditional science disciplines where research and practice are distinct, in HCI the two intermingle. In particular, this leads to theoretical discussions of HCI methodology (see Chapter XXX) and design practice, such as the task–artefact cycle (Carroll and Rosson, 1992).

Several of the terms used in this chapter are ones that I have coined at different times to describe concepts or techniques that may be widely practised, but did not have a name. These include synthetic theory in section 1.1.4, critical transitions and recoding dialectic in section 1.3.2, and generative artefacts in section 1.5.3. More detail on these and other aspects of research methods in HCI can be found on my *Research and Innovation Techniques* web pages (Dix, 1996–2007).

References

Card, S., Moran, T. and Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates.

Carroll, J. and Rosson, M. (1992). Getting around the task-artifact cycle: how to make claims and design by scenario. *ACM Trans. Inf. Syst*. 10(2):181–212.

Carroll, J., Ed. (2003). *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science.* (Interactive Technologies), Morgan Kaufmann.

Dix, A. (1996–2007). Research and Innovation Techniques. http://www.hiraeth.com/alan/topics/res-tech/

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

Dix, A. (2002). Teaching Innovation Keynote at *Excellence in Education and Training Convention*, Singapore Polytechnic, May 2002.

Dix, A. (2007). Why examples are hard, and what to do about it. *Interfaces*, 72:16–18. Autumn 2007. http://www.hcibook.com/alan/papers/ examples-2007/

Ellis, G. and Dix, A. (2006). An explorative analysis of user evaluation studies in information visualisation. In *Proceedings of the 2006 Conference on Beyond Time and Errors: Novel Evaluation Methods for Information Visualization* (Venice, Italy, May 23, 2006). BELIV '06. ACM Press, New York, NY, pp. 1-7.

Fawcett, J. and Downs, F. (1986). *The Relationship of Theory and Research*. Norwalk, CT: Appleton Century Crofts.

Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391. (Reprinted in *Journal of Experimental Psychology: General*, 121(3):262--269, 1992).

Gardner, H. (1983). *Frames of Mind: The Theory of Multiple Intelligences*. New York: Basic Books.

Glaser, B. and Strauss, A. (1967). *The discovery of grounded theory*. Chicago: Aldine.

Heritage, J. (1984). *Garfinkle and Ethnomethodology*. Polity Press, Cambridge, Mass, USA.

Keele. S. (1968). Movement Control in Skilled Motor Performance. *Psychological Bulletin*. 70:387–402.

Kuhn, T. (1962/1970). *The Structure of Scientific Revolutions*. Chicago: University of Chicago Press (1962, 1st edition; 1970, 2nd edition, with postscript).

Long, J. and Dowell, J. (1989). Conceptions of the Discipline of HCI: Craft, Applied Science, and Engineering. In: Sutcliffe, Alistair and Macauley, Linda (eds.) *Proceedings of the HCI89 - People and Computers V*, Cambridge University Press. pp. 9-32.

Meyer, D., Abrams, R., Kornblum, S., Wright, C., and Smith, J. (1988). Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review*, 95:340–370.

Minsky, M. (1965). Matter, Mind and Models. *Proc. International Federation of Information Processing Congress 1965*, Volume 1, pp. 45–49. http://web.media.mit.edu/~minsky/papers/MatterMindModels.html

OED (1973). Theory (definition). *The Shorter Oxford English Dictionary on Historical Principles. Third Edition.* Volume II. W. Little, H. Fowler, J. Coulson C. Onions, G. Friedrichsen (eds.), Clarendon Press, Oxford, pp. 2281.

Plain English Campaign (2003). *Foot in Mouth Award 2003*. (accessed 26 July 2007). http://www.plainenglish.co.uk/footinmouth.htm#2003

Popper, K. (1959). *The Logic of Scientific Discovery*. Basic Books, New York, NY, 1959.

Suchman, L. (1987). *Plans and Situated Actions: the problem of human-machine communication*. Cambridge University Press.

Winograd, T. and Flores, F. (1985). *Understanding Computers and Cognition*. Ablex Publishing Corp.