CHAPTER 4
Classical Theories

In the early 1980s, there was much optimism as to how the field of cognitive psychology could significantly contribute to the development of the field of HCI. A driving force was the realization that most computer systems being developed at the time were difficult to learn, difficult to use and did not enable the users to carry out the tasks in the way they wanted. The body of knowledge, research findings and methods that made up cognitive psychology were seen as providing the means by which to reverse this trend, by being able to inform the design of easy to learn and use computer systems. Much research was carried out to achieve this goal: mainstream information processing theories and models were used as a basis from which to develop design principles, methods, analytic tools and prescriptive advice for the design of computer interfaces (e.g., Carroll, 1991). These are loosely classified into three main approaches: body of knowledge, applying basic research and cognitive modeling (Rogers, 2004). Under the heading cognitive modeling, well-known early conceptual modeling approaches are outlined, including the interface gulfs, GOMS and mental models.

4.1 BODY OF KNOWLEDGE

The most widely known contribution that the field of cognitive psychology made to HCI is the provision of explanations of the capabilities and limitations of users, in terms of what they can and cannot do when performing computer-based tasks. For example, theories that were developed to address key areas, like memory, attention, perception, learning, mental models and decision-making, have been much popularized in tutorials, introductory chapters, articles in magazines and the web to show their relevance to HCI. Popularized examples of this approach included Norman (1988), Preece et al. (1994) and Monk (1984). By explicating user performance in terms of well-known cognitive characteristics that are easy to assimilate (e.g., recognition is better than recall), designers were alerted to their possible effects when making design decisions — something that they might not have otherwise considered.

A well-known example is the application of the finding that people find it easier to recognize things shown to them than to have to recall them from memory. Most graphical interfaces have been designed to provide visual ways of presenting information, that enable the user to scan and recognize an item like a command, rather than require them to recall what command to issue next at the interface.

This approach, however, has tended to be piecemeal — depending on the availability of research findings in cognitive psychology that can be translated into a digestible form. A further problem with this approach is its propensity towards a "jewel in the mud" culture, whereby a single research finding sticks out from the others and is much cited, at the expense of all the other results (Green et al., 1996). In HCI, the "magical number 7+−2" (George Miller's theory about memory, which is that only 7+−2 chunks of information, such as words or numbers, can ever be held in short-term memory at any one time) became the de facto example: nearly every designer had heard of it but not necessarily where it had come from or what situations it is appropriate to apply. A consequence was that it largely devolved into a kind of catchphrase, open to interpretation in all sorts of ways, which ended up being far removed from the original idea underlying the research finding. For example, some designers interpreted the magic number 7+−2 to mean that displays should have no more than 7+−2 of a category (e.g., number of colors, number of icons on a menu bar, number of tabs at the top of a web page and number of bullets in list), regardless of context or task, which is clearly in many cases inappropriate (Bailey, 2000).

4.2 APPLYING BASIC RESEARCH

A more systematic approach was to select relevant cognitive theories that could be applied to interface design concerns. For example, theories about human memory were used to decide what was the best set of icons or command names to use, given people's memory limitations. One of the main benefits of this approach was to help researchers identify relevant cognitive factors (e.g., categorization strategies, learning methods, perceptual processes) that are important to consider in the design and evaluation of different kinds of GUIs and speech recognition systems. It can also help us understand new kinds of computer-augmented behaviors by examining human's abilities and limitations when interacting with technologies. In particular, the theories demonstrate what humans are good and bad at and, based on this knowledge, can inform the design of technologies that both extend human capabilities and compensate for their weaknesses.

A core lesson that was learned, however, is that you cannot simply lift theories out of an established field (i.e., cognitive psychology), that have been developed to explain specific phenomena about cognition, and then reapply them to explain other kinds of seemingly related phenomena in a different domain (i.e., interacting with computers). This is because the kinds of cognitive processes that are studied in basic research are quite different from what happens in the "real" world of human-computer interactions (Landauer, 1991). In basic research settings, behavior is controlled in a laboratory in an attempt to determine the effects of singled out cognitive processes (e.g., short-term memory span). The processes are studied in isolation and subjects (sic) are asked to perform a specific task, without any distractions or aids at hand. In contrast, the cognition that happens during human-computer interaction is much more messy, whereby many interdependent processes are involved for any given activity. Moreover, in their everyday and work settings, people rarely perform a task in isolation. Instead, they are constantly interrupted or interrupt their own activities, by talking to others, taking breaks, starting new activities, resuming others, and so on. The stark differences between a controlled lab setting and the messy real world setting, meant that many of the theories derived from the former were not applicable to the latter. Predictions based on basic cognitive theories about what kinds of interfaces would be easiest to learn, most memorable, easiest to recognize and so on, were often not supported.

The problem of applying basic research in a real world context is exemplified by the early efforts of a number of cognitive psychologists in the early 1980s, who were interested in finding out what was the most effective set of command names for text editing systems, in terms of being easy to learn and remember. At the time, it was a well-known problem that many users and some programmers had a difficult time remembering the names used in command sets for text editing applications. Several psychologists assumed that research findings on paired-associate learning could be usefully applied to help overcome this problem; this being a well developed area in the basic psychological literature. One of the main findings to be applied was pairs of words are learned more quickly and remembered if subjects have prior knowledge of them (i.e., highly familiar and salient words). It was further suggested that command names be designed to include specific names that have some natural link with the underlying referents they were to be associated with. Based on these hypotheses, a number of experiments were carried out, where users had to learn different sets of command names, that were selected based on their specificity, familiarity, etc. The findings from the studies, however, were inconclusive; some found specific names were better remembered than general terms (Barnard et al., 1982), others showed names selected by users, themselves, were preferable (e.g., Ledgard et al., 1981; Scapin, 1981) while others demonstrated that high-frequency words were better remembered than low-frequency ones (Gunther et al., 1986). Hence, instead of the outcome of the research on command names being able to provide a generalizable design rule about which names are the most effective to learn and remember, it suggested that a whole range of different factors affects the learnability and memorability of command names. As such, the original theory about naming was not able be applied effectively to the selection of optimal names in the context of computer interfaces.

4.3 COGNITIVE MODELING

Another approach that was developed was to model the cognition that is assumed to happen when a user carries out their tasks. Some of the earliest models focused on user's goals and how they could achieve (or not) them with a particular computational system. Most influential at the time were Shneiderman's (1983) framework of direct manipulation (that explicated the properties of graphical user interfaces in terms of continuous representation of objects of interest, and rapid, reversible, incremental actions and feedback enabling a user to directly manipulate objects on the screen, using actions that loosely correspond to those in the physical world), Hutchins et al.'s (1986) conceptual framework of directness (that describes the gap between the user's goals and the way a system works in terms of gulfs of execution and evaluation), and Norman's (1986) theory of action (that models the putative mental and physical stages involved in carrying out an action when using a system). The latter two were heavily influenced by contemporary cognitive science theory of the time, which itself, focused on modeling people's goals and how they were met.

Interface Gulfs in a Nutshell

The idea of there being gulfs at the interface that needed to be bridged was influential in the early theorizing of HCI. Essentially, the gulf of execution and the gulf of evaluation describe the gaps that exist between the user and the interface (Hutchins et al., 1986). They are intended to show how to design the latter to enable the user to cope with them. The first one — the gulf of execution — describes the distance from the user to the physical system while the second one — the gulf of evaluation — is the distance from the physical system to the user (see Figure 4.1). It was proposed that designers and users needed to concern themselves with how to bridge the gulfs in order to reduce the cognitive effort required to perform a task. This could be achieved, on the one hand, by designing usable interfaces that match the psychological characteristics of the user, e.g., taking into account their memory limitations, and, on the other hand, by the user learning to create goals, plans and action sequences that fit with how the interface works.

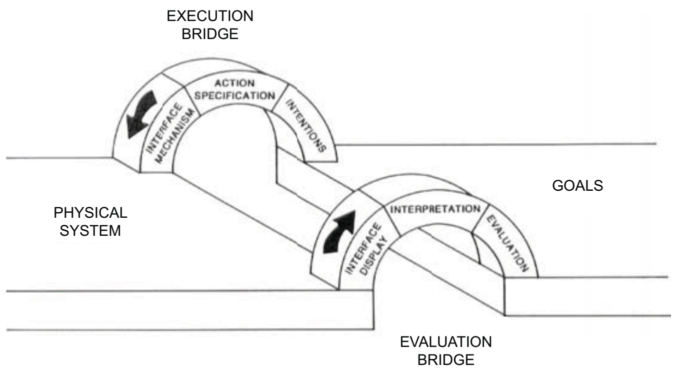

Figure 4.1: Bridging the gulfs of execution and evaluation (from Rogers et al., 2011).

Norman and Hutchins et al.' s respective early cognitive models of the user provided heuristics by which to conceptualize and understand the interactions that were assumed to take place between a user and a system. In so doing, they suggested new ways of thinking about designing interfaces, such as GUIs. In contrast, Card et al.'s (1983) cognitive model of the user, called the model human processor, aimed to be more scientific, by providing a basis from which to make quantitative predictions about user performance and, in so doing, a way of enabling researchers and developers to evaluate different kinds of interfaces in terms of their suitability for supporting various tasks (see Figure 4.2).

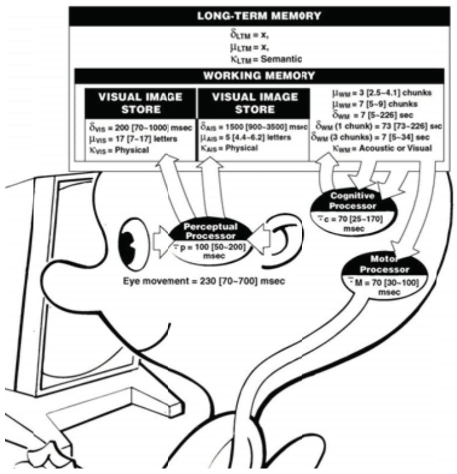
LONG-TERM MEMORY
δ_{LTM} = x,
μ_{LTM} = x,
κ_{LTM} = Semantic
WORKING MEMORY
VISUAL IMAGE STORE
δ_{VIS} = 200 [70–1000] msec
μ_{VIS} = 17 [7–17] letters
κ_{VIS} = Physical
VISUAL IMAGE STORE
δ_{AIS} = 1500 [900–3500] msec
μ_{AIS} = 5 [4.4–6.2] letters
κ_{AIS} = Physical
μ_{WM} = 3 [2.5–4.1] chunks
μ_{WM} = 7 [5–9] chunks
δ_{WM} = 7 [5–226] sec
δ_{WM} (1 chunk) = 73 [73–226] sec
δ_{WM} (3 chunks) = 7 [5–34] sec
κ_{WM} = Acoustic or Visual
Cognitive Processor
τ_c = 70 [25–170] msec
Perceptual Processor
τ_p = 100 [50–200] msec
Eye movement = 230 [70–700] msec
Motor Processor
τ_M = 70 [30–100] msec

Figure 4.2: The Model Human Processor.

GOMS in a Nutshell

Based upon the established information processing model of the time, Card et al.'s (1983) developed a model of the user, called the model human processor (MHP). It comprised three interacting systems: perceptual, cognitive and motor, each with their own memory and processor. To show how the model could be used to evaluate interactive systems, Card et al. developed a set of predictive models, collectively referred to as GOMS (Goals, Operators, Methods and Selection rules). The resulting suite of methods provided usability engineers with descriptive and analytic tools that were befitting of the engineering approach that was dominant in HCI at the time.

For a while, during the late 1980s and early 1990s, GOMS proved to be highly popular. It was the staple of many courses in HCI and interactive design that started to be run as part of the undergraduate Computer Science curriculum in the U.S. and Europe. Case studies were reported about its success for comparing the efficacy of different computer-based systems (see Olson and Olson, 1991).

The most well-known GOMS success story was Project Ernestine, where a group of researchers carried out a GOMS analysis for a modern workstation that a large phone company were contemplating purchasing, and counter-intuitively, predicted that it would perform worse than the existing computer system being used at the company, for the same kind of tasks. A consequence was that they advised the company not to invest in what could have been potentially a very costly and inefficient technology (Atwood et al., 1996).

While this study has shown that the GOMS approach was useful in helping make purchasing decisions about the effectiveness of new products (although, obviously, they were unable to validate whether if the company had invested in them, it would have been detrimental), it did not take off as a widely used evaluation method. This was due partly to it only being able to model reliably computer-based tasks that involve a small set of highly routine data-entry type tasks. Furthermore, it is intended to be used to predict expert performance, and does not allow for errors to be modeled. This makes it much more difficult (and sometimes impossible) to predict how most users will carry out their tasks when using systems in their work, especially those that have been designed to be flexible in the way they can be used. In most situations the majority of users are highly variable in how they use systems, often carrying out their activities in quite different ways to that modeled or predicted. Many unpredictable factors come into play. These include individual differences among user's, fatigue, mental workload, learning effects and social and organizational factors (Olson and Olson, 1991). Moreover, most people do not carry out their tasks sequentially but tend to be constantly multitasking, dealing with interruptions and talking to others, while carrying out a range of activities.

A problem with using these kinds of predictive models, therefore, is that they can only make predictions about isolated predictable behavior. Given that most people are often unpredictable in the way they behave and, moreover, interweave their ongoing activities in response to unpredictable external demands, it means that the outcome of a GOMS analysis can only ever be a rough approximation.

There have been a number of efforts to make GOMS more usable and applicable to a range of interfaces. Bonnie John has been one of its chief proponents, tirelessly publishing and adapting it for different uses, including CPM-GOMS (John and Gray, 1995) that can model multitasking behavior and CogTool (Teo and John, 2008) that enables non-psychologists to create cognitive models of user tasks from which reliable estimates of skilled user task times can be derived. A number of other cognitive models were developed post GOMS, aimed at predicting user behavior when using various kinds of systems (e.g., the EPIC model, Kieras and Meyer, 1997). Similar to the various versions of GOMS, they can predict simple kinds of user interaction fairly accurately, but are unable to cope with more complex situations, where the amount of judgment a researcher or designer has to make, as to which aspects to model and how to do this, greatly increases (Sutcliffe, 2000). The process becomes increasingly subjective and involves considerable effort, making it more difficult to use them to make predictions that match the ongoing state of affairs.

In contrast, cognitive modeling approaches, that do not have a predictive element to them, have had a wider and more sustained use. Examples include heuristic evaluation (Mohlich and Nielsen,

1990) and cognitive walkthroughs (Polson et al., 1992) that are more widely used by practitioners. Such methods provide various heuristics and questions for evaluators to operationalize and answer, respectively. An example of a well-known heuristic is "minimize user memory load." As such, these more pragmatic methods differ from the other kinds of cognitive modeling techniques insofar as they provide prescriptive advice that is largely based on assumptions about the kinds of cognitive activities users engage in when interacting with a given system.

Mental Models in a Nutshell

Another popular concept that many researchers were attracted by to account for human-computer interactions, but ultimately struggled to explicate adequately, was mental models. The reason the term was so appealing was that it suggested a more dynamic way of characterizing the knowledge that people are assumed to have when interacting with a system, how that enables them to understand how a system works and to know what to do next. However, despite many attempts to capture and unpack the notion of mental models (see Rogers et al., 1992) it was difficult to show they existed and "ran" in the mind as assumed. Just as it has proved to be problematic in cognitive science, to infer the kinds of representations used in the mind it was, likewise, impossible to say with any confidence whether people really developed mental models and if they did, how they functioned in the mind.

In their original instantiation, mental models were postulated as internal constructions of some aspect of the external world that are manipulated, enabling predictions and inferences to be made (Craik, 1943). The process was thought to involve the fleshing out and the running of a mental model (Johnson-Laird, 1983), involving both unconscious and conscious mental processes, where images and analogies were activated. It was argued that the more someone learns about a system and how it functions, the more their mental model develops. For example, appliance engineers have a deep mental model of how home entertainment systems work that allows them to work out how to set them up and fix them. In contrast, an average citizen is likely to have a reasonably good mental model of how to operate their home entertainment system but a shallow mental model of how it works.

Hence, mental models seemed an obvious concept to import into HCI, where a goal was to account for, and, ultimately model how users understand the working of computer systems. An assumption was that if we could tap into this knowledge, we would be able to predict how well people could reason about an interactive system, and provide the necessary training materials, interface types, etc., that would enable them to learn how to use it effectively and know how to troubleshoot if something went wrong at the interface or they made a mistake, and needed to work out how to rectify it.

A number of studies were conducted; people were asked about their knowledge and also observed using devices; many concluded that people's understanding of how computer-based technologies and services, e.g., the Internet, wireless networking, broadband, search engines and viruses, work was alarmingly poor. Norman's earlier assertion (Norman, 1983) that people's mental models are often incomplete, easily confusable, based on inappropriate analogies and superstition was found to be the case. It was discovered that many find it difficult to identify, describe or solve a problem, and lack the words or concepts to explain what is happening.

It was assumed that the situation could be improved if we could find ways of helping users to develop better mental models of interactive systems. Ideally, they should be able to develop a mental model that matched the designer's conceptual model of how they had envisioned the system working. Making the system image more transparent for the users so they could see how it worked and educating them more about how it worked were the two main prescriptive strategies.

The term mental model largely fell into disuse towards the end of the 1990s, as newer theories came to the fore that proposed alternative ways of how people interacted with and understood technologies. These theories moved away from exclusively accounting for the assumed representations inside the head of a user towards explicating a bigger picture of how cognition worked in the world; theories of embodied, distributed and situated action were proposed for how people interacted with the world, with a focus on the artifacts and external representations they use and generate. A major thrust was to unpack interactivity *per se* rather than capturing a putative internal model (Kirsh, 1997); this was more observable and more in line with what users did when using computers. They don't close their eyes to run an internal model in their minds and then open their eyes and apply it to the task in hand. The internal representations that are activated are used in conjunction with the many different forms of external representations, coupled with an array of physical and mental actions, including gesturing, projecting, talking, touching, manipulating, and imagining. As described in the next section, the newer theories provided more scope for inspection and validation — to understand how people make sense of the world around them and the technology they encounter and try to master — than the earlier, more elusive theories of mental models. They also offer more potential for developing generative tools, suggesting how to design interfaces that optimize certain kinds of desired interactivity.

What impact has cognitive modeling had in HCI?

Cognitive modeling approaches made a big impact in the late 1980s and early 1990s, providing techniques for predicting and analyzing users cognitive tasks. The early GOMS family showed how cognitive models could be adapted in HCI. However, models are only ever simplifications of real behavior, and as such, are limited in how and what they capture. So much depends on the researcher's operationalization of human knowledge in terms of putative kinds of internal representations and processes. While these approximations and characterizations enabled them to compare different ways users might perform problem-solving tasks, such as word processing using different operations or searching for information on the web using different browsers, they waned in use in the 1990s when it became more widely recognized that the context of use and external resources and representations were an integral part of the usability of different interfaces.