Cognitive Design Computing

John S. Gero[1]
[1]*Department of Computer Science and School of Architecture, University of North Carolina at Charlotte and Krasnow Institute for Advanced Study, George Mason University*
[1]*john@johngero.com*

This talk describes the foundational concepts of cognitive design computing and then presents some examples. Cognitive computing is concerned with modeling human cognition computationally and using that model as the foundation for constructing computer models of design activities. Human cognition is based on perception, learning and adaptation. Here we present human cognition in terms of situated cognition - cognition involving interaction with an environment. The talk briefly introduces a set of principles for cognitive design computing founded on the three concepts of interaction, constructive memory and situatedness. It then presents two examples of applications of this approach.

INTRODUCTION

Computational design tools aim at encoding knowledge and making it available in an objective manner to the designer. There is an assumption behind this approach, namely that all the knowledge is objective, ie, independent of the user. Examples of objective knowledge include stress analysis, determining the position of the sun, thermal analysis of a building and methods such as linear programming in optimisation. Such objective knowledge tends to be deductive in nature. The most powerful examples of deductive knowledge are those based on axioms from which subsequent theorems have been developed. These theorems map onto the behaviour of the world. In addition, there is a category of knowledge that is based on induction (ie, knowledge learned from examples without causality). Even inductive knowledge is treated as objective, ie, independent of the user. For example, a layout algorithm that utilises some heuristics is used as if there were causality encoded. This has served design computing well. It has allowed for the widespread distribution of computational tools. It has provided the basis of transferable skill development in users. As our knowledge of the world has improved, so we have

been able to update and adapt the knowledge in these programs.

This talk presents an approach that aims to extend our understanding of what kinds of knowledge we can expect our computational tools to have and how systems that have a range of kinds of knowledge might perform differently.

FIRST-PERSON VERSUS THIRD-PERSON KNOWLEDGE

We call such objective knowledge *third-person knowledge* in that the person who produced the knowledge is not required to be there when that knowledge is used by another person. For example, even though Newton is dead his laws continue to work fine. Whilst it is clear that much of human knowledge is of this third-person kind, in the sense described above, there is a category of everyday knowledge that depends on the person rather than deduction. This kind of knowledge develops through the interaction of the individual and their world and as a consequence is personal knowledge. It is called *first-person knowledge*. This class of knowledge is sometimes inappropriately encoded as third person knowledge and when done so often causes the mismatch between the experience of the person who coded the knowledge and a subsequent user of that knowledge.

A simple example of such encoding of personal knowledge can be seen even in the way objects are represented in a CAD software system. Figure 1(a) shows the screen image of a floor layout. Simply looking at the drawing of the floor layout gives no indication of how it has been encoded. The darkened line is the single polyline representation of the outline obtained by pointing to a spot on the boundary, but that representation could not be discerned from the image. Figure 1(b) shows exactly the same outline but it is encoded differently, as indicated by the darkened polyline obtained by pointing to the same spot.

The issue here is one of interpretation that is often missing in computation. A common assumption is that the external (and even the internal) world is there to be represented, ie, that in some sense it has only one representation. This misses an important step: namely that of interpretation, which depends on the viewer not on the underlying external representation. Before anything can be represented it needs first to be interpreted and it is this interpretation that is represented. This is an example of first-person interaction with the external world that results in first-person knowledge about the world.

How can we build computation systems that encode first-person as well as third-person knowledge? To do this we rely on concepts from cognitive science and in particular a branch called *situated cognition*. Using those concepts we can produce a branch of computing called *cognitive computing* that is a closer analog to how the mind works than general computing.

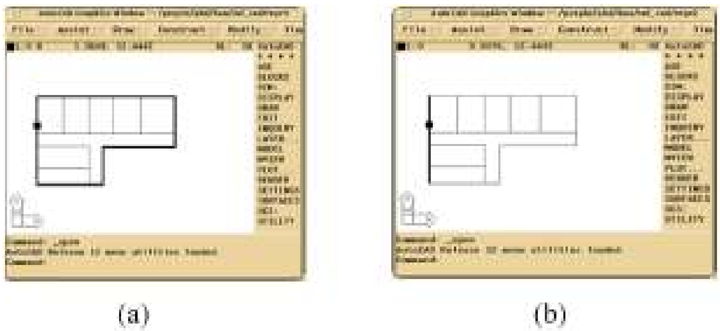

Figure 1
The same image has different encodings (a) and (b) that depend on the individuals who created them rather than on any objective knowledge.

SITUATED COGNITION

Cognitive science is concerned with understanding and representing structures and processes in the mind (as opposed to brain science which is concerned with understanding and modeling structures and processes in the brain). Situated cognition is concerned cognition that is embodied and a consequence of embodiment is an increased focus on the interaction between the computational system and its environment (Clancey, 1997; 1999; Gero, 1998; Suchman, 1987).

Situated cognition is founded on three ideas:

1. Interaction: knowledge comes from encoding and through interaction; in particular first-person interactions with representations, which includes the expectations of the person or system carrying out the interpretation (Agre, 1997; Smith and Gero, 1998; Zhang, 1997);
2. Constructive memory: constructive memory, which is concerned with memory as a process to generate a memory cued on a demand to have such a memory rather than a recall of elements in a location (Bartlett, 1932/1977; Dewey, 1896; Gero, 1999; Rosenfeld, 1988; von Glaserfeld, 1995);
3. Situatedness: situatedness is concerned with being in a particular place at a particular time and how the world is viewed by each individual from that place at that time (Suchman, 1987).

PRINCIPLES FOR COGNITIVE DESIGN COMPUTING

We can develop a set of principles that form the foundation for the development of cognitive design computing.

Principle of Effect: What you do matters

The implication of this principle is that actions produce effects in terms of the production of first-person knowledge. In traditional uses of computers in designing, the computational systems are unchanged by their use. This makes sense if the system only embodies third-person knowledge. However, the use of any system, whatever kind of knowledge it contains, generates first-person knowledge in the system that uses it. This is the first fundamental distinction between traditional design computing and situated design computing. For example, an optimization approach used in the design of layouts could produce first-person knowledge in the form successful strategies. These could then be used next time the optimization program was used for layouts.

Principle of Ordered Temporality: When you do what you do matters

In traditional design computing when you carry out a computation plays no role in the result that is produced. Again, this makes sense if the system only embodies third-person knowledge. However, if the system embodies first-person and generates first-person-knowledge as it runs, the chronology of the system's use affects what is used and what is learned. As the boundary condition for this principle consider that if A is carried out before B, then A cannot make use of any knowledge acquired through the execution of B, but B can make of knowledge acquired through the execution of A.

Lemma of Experience: What you did before affects what you do now

As a consequence of principles 1 and 2, we can state this lemma: What a system did before affects it does now. This is one definition of experience. Experience has the potential to guide future actions. The effect of this is that cognitive design computing systems are not static systems but are dynamic in terms of their behavior. This concept can be applied recursively, so that previous experiences are used to produce current experiences.

Principle of Locality: Where you are when you do what you do matters

First-person knowledge includes not only what happened and when it happened but also where it happened. The system has to be at the ”right” place at the ’right” time for unique events to be interpreted.

Principle of Interaction: Who and what you interact with matters
Interaction is one of the distinguishing characteristics of a cognitive design computing system. Without interaction there is no potential to produce first-person knowledge. There are two sources of interaction for a system: other computer programs and users of the system. Conceptually there is no difference between them in terms of interactions. Each interaction has the capacity to use previous experiences and to produce first-person knowledge. Interactions have the potential to change the meanings of experiences.

Principle of Ontology: What you think the world is about affects what it is about for you.
This implies that cognitive design computing develops a representation of the situation that provides the basis for any interpretation of what is observed.

ACKNOWLEDGEMENTS

The research reported here has been funded by grants from the US National Science Foundation Grant numbers CNS-0745390, SBE-0915482 and CMMI-1400466.

REFERENCES

Agre, PE 1997, *Computation and Human Experience*, Cambridge University Press, Cambridge
Bartlett, FC 1932/1977, *Remembering: A Study in Experimental and Social Psychology*, Cambridge University Press, Cambridge
Clancey, W 1997, *Situated Cognition*, Cambridge University Press
Clancey, W 1999, *Conceptual Coordination*, Cambridge University Press
Dewey, J 1896, ’The reflex arc concept in psychology’, *Psychological Review*, 3, pp. 357-370
Gero, JS 1999, ’Constructive memory in design thinking’, in Goldschmidt, G and Porter, W (eds) 1999, *Design Thinking Research Symposium: Design Representation*, MIT, Cambridge, pp. I.29-35
Gero, JS 1998, ‘Conceptual designing as a sequence of situated acts’, in Smith, I (eds) 1998, *Artificial Intelligence in Structural Engineering*, Springer, Berlin, pp. 165-177
von Glasersfeld, E 1995, *Radical Constructivism: A Way of Knowing and Learning*, The Falmer Press
Rosenfield, I 1988, *The Invention of Memory*, Basic Books, New York
Smith, G and Gero, JS 2002, ’Interaction and experience: Situated agents and sketching’, in Gero, JS and Brazier, F (eds) 2002, *Agents in Design 2002*, Key Centre of Design Computing and Cognition, University of Sydney, Australia, pp. 115-132
Suchman, L 1987, *Plans and Situated Actions: The Problem of Human-Machine Communication*, Cambridge University Press
Zhang, J 1997, ’The nature of external representations in problem-solving’, *Cognitive Science*, 21, pp. 179-217