SITUATED COMPUTING

A New Paradigm for Design Computing

JOHN S. GERO
Key Centre of Design Computing and Cognition
University of Sydney
NSW 2006 Australia
Email address: john@arch.usyd.edu.au

Abstract. This paper introduces the concepts of situatedness and constructive memory as the foundations of situated computing. The difference is between encoding all knowledge prior to its use and allowing the knowledge to be developed and grounded in the interaction between the external world and the designer/tool. The paper elaborates these concepts and concludes with a discussion of the implication of situated computing on computational models of designing and on the development of adaptive design tools.

1. Introduction

Design computing involves all facets of the use of computers to support the acts of designing. Much of the research in design computing has been focused on modeling and representing designed objects, whether for the individual designer or for teams of designers using the World Wide Web. There has continued to be an interest in computational aids that support designing more directly and in producing different design environments. Amongst the former are evolutionary systems and amongst the latter are virtual environments. All of this work is based a paradigm of computing that assumes that the underlying programs are unchanged by their use and certainly the underlying programs are not affected by where or how they are used. This is one of the foundations of objective knowledge: it is independent of who uses and what is done with it. We expect this of our objective knowledge. Objective knowledge is knowledge about the structure of the world. Thus, we expect that an object in a CAD model is unchanged by the view we present of it; that the algorithm that calculates shadows is unchanged by whomever uses it; and that the evolutionary algorithm used in a layout program is unchanged by using it on different types of problems.

I do not wish to argue against this idea but rather to suggest that this is not a complete characterization of the use of our computational tools. In order for a human or a computer to make use of a tool they require other sorts of knowledge than simply what makes the tool work and how it works. I would suggest that many times the tool user does not know how the tool works only how to use the tool. Thus, knowledge about a tool may be thought of as comprising two layers: knowledge about the tool and knowledge about the use of the tool. If we build computational tools it may not be sufficient to simply encode objective knowledge in them, we should think of encoding some of the tool use knowledge in the tool.

Some researchers have suggested that our computational tools should learn as they are being used in order to keep pace with the user's increase in experience in using them (Gero 1996). This paper presents a new paradigm for design computing called situated computing that, it is claimed, follows designing more closely and has the potential to change how we build our tools so that they may be more active and hence more useful.

2. Situated Computing

Situated computing makes use of concepts from situated cognition (Clancey 1997), which holds that where are you and when you are there matters and that the state you are in affects what you do. The fundamental difference is between encoding all knowledge prior to its use and allowing the knowledge to be developed and grounded in the interaction between the tool and its environment. The effect of this is to provide a computational system such as a tool with experience based on its interaction with its environment. That experience is then used to guide its future actions. The effect of this grounded experience is to provide the tool with the capability to respond differently when exposed to the same environment again depending on the experiences it had in between the two exposures. The objective knowledge within the tool is unchanged, only the knowledge that is the result of the interaction of the tool with its environment is changed. It provides the basis for computational systems to learn and change their behaviour based on their experiences. The learning is not necessarily to improve the performance of the system rather it is to customize it to its user.

Situated computing is a computational paradigm founded on the two concepts of situatedness and constructive memory.

2.1. SITUATEDNESS

Situatedness embodies the concept that “where you are when you do what you do matters”. The situation is the designer’s or the tool’s interpretation of its environment and itself in that environment at that time. Let me present a number of examples from our shared experiences that illustrate this point.

2.1.1. Where you are when, matters

Take as an example the images in Figures 1(a) and 1(b). You could see each of these individually at different times but it is not until you see both of them at the same time in the same place that you can see the emergent figure of a vase that appears in Figure 1(c): where you are when, matters. This has implications for the way in which appropriate tools may be selected or the way in which a tool may have to configure itself as the result of emergence is a shift of focus and representation.

(a) (b) (c)

Figure 1. Where you are when matters.

2.1.2. What you are looking for affects what you see

Unlike the notion of objective knowledge where the object of the application of the knowledge is unaffected by the knowledge, what you are looking for affects what you see. Take as an example the image in Figure 2(a). Its meaning is unclear by itself. However, if we are looking for the image of a letter to fill out the word “cat” then it looks like an “A”. If, however, we are llokng to fill out the word “the” then it looks like the letter “H”.

(a) (b) (c)

Figure 2. What you are looking for affects what you see.

This has important implications for our understanding of what sorts of digital representations we use. One implication is that the representation should not be fixed a priori otherwise the example above will not be possible.

Figure 3 presents another example of this idea that the representation of an object depends not only on the object but also on what you “see”. Is this a set of triangles pointing rightwards, with their bases facing left? Or is it a set of triangles pointing upwards to the left with their bases facing downwards and to the right? Or is it a set of triangles pointing downwards to the left with their bases facing upwards to the right? There is no “right” answer. It depends on what you see.

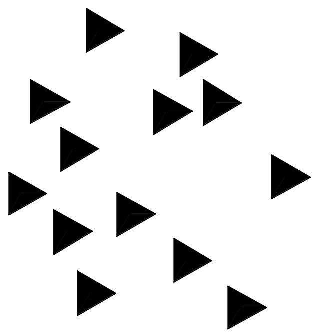

Figure 3. There is no fixed representation of the world: the representation depends of what you “see”.

Situatedness can be a slippery concept but its distinguishing features include the interaction between a tool and its environment and the notion that what is “seen” is not only a function of what is “out there” but also a function of what is inside the tool.

This brings us to another of the foundational concepts of situated computing, namely that of constructive memory.

2.2. CONSTRUCTIVE MEMORY

We are accustomed to thinking of memory as a thing in a location. So that when a person is asked their telephone number we assume that they are recalling it in some sense in the same way that a computer does by indexing it. Computers can access their memory in a number of ways (by an index, by a location, or by its content) but the memory is unchanged by its recall. It is suggested that this form of memory is a limited view of memory (Rosenfield 1988) and does not address a fundamental attribute of memory which is that is not composed merely of facts that can be recalled but rather is a process of construction. Most "memories" do not exist waiting to be recalled but are constructed on demand – the demand to have a memory. We call the first kind of memories "fact" or "experience" memories and the second kind "constructive" memories or just memories.

Let's imagine that we are looking at a computational agent (ie a computer program that has the capability to exhibit some autonomy in its decision making) and examine the following where the agent has some sensate experiences that produce fact memories. Figure 4 shows the case where a memory is required by the situation and the agent constructs a memory from the experience it has had, its knowledge and the situation it finds itself in. This new memory did not exist before and is a function of the experience memory, the agent's knowledge and the request from the situation. A different request would have produced a different memory. If the agent has new experiences the same question will not necessarily produce the same memory as the new experience may participate in the construction of the new memory that is then likely to be different. This is one of the important distinguishing features of constructive memory – the ability to incorporate new experiences into the construction of new and different memories.

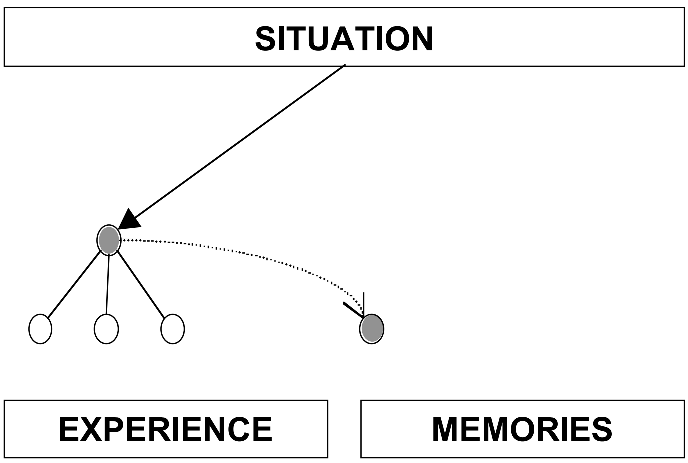

Figure 4. New memories are constructed in response to a demand to have such a memory.

When a new request for a memory is made it may not be directly affected by any memory and may depend only on previous experiences, Figure 5. However, since each new memory may be part of the situation the agent is in, it may indirectly affect the newly constructed memory. Such new memories are still constructed memories. These types of constructed memories are rare. In general all new memories are directly affected by previous memories as well as experiences.

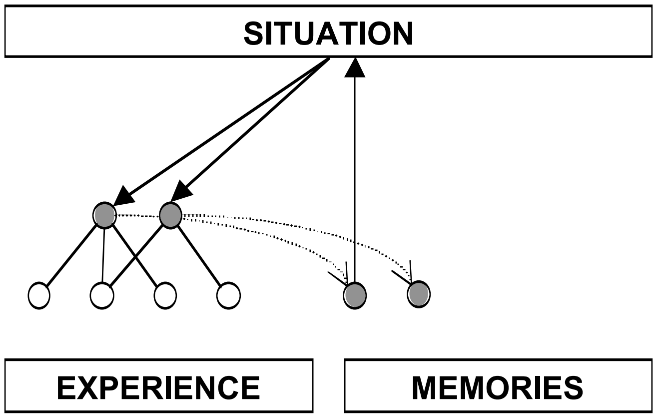

Figure 5. Additional demands for new memories are not necessarily affected by earlier constructed memories but may include new experiences, experiences that were not there when a previous memory was constructed.

Figure 6 shows a new, constructed memory whose result is directly affected by a previously constructed memory as well as being indirectly affected by previously constructed memories (the upward pointing arrows that form part of the situation) as well as new experiences.

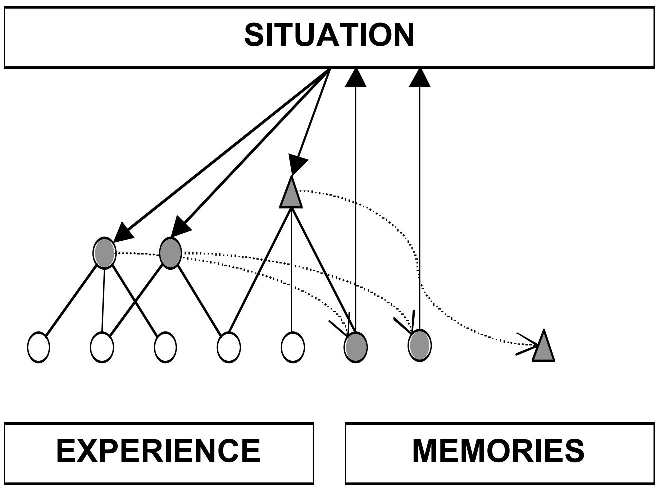

Figure 6. Additional demands for new memories are normally affected by earlier constructed memories.

Figure 6 illustrates the range of inputs and processes involved in constructive memory. A new memory can be constructed using any of:

- situation
- old experiences
- new experiences
- previously constructed memories.

Constructive memory can be implemented in a variety of ways. Current approaches make use of neural networks (Saunders and Gero 2001; Liew and Gero 2002) and rule-based systems (Maher, Smith and Gero 2003).

3. Situated Design Computing

Computational models of designing have largely been founded on fixed views of the world, often derived using artificial intelligence models (Gero, 1990; Gero, 2000; Gero, 2003). Whilst these models have been useful they have proven to be inadequate to describe much of the detailed behaviour of designers observed in protocol studies (Schön and Wiggins, 1992). This behaviour can be more readily modeled using situated computing and forms the basis of situated design computing – the inclusion of situated concepts into design computing.

In order to include situatedness we need to reconceptualise what is happening in any computational model of designing. Gero and Kannengiesser (2002) have taken the Function–Behaviour–Structure model of designing and introduced situatedness into it through the use of a three-worlds model derived from a cognitive view of designing.

The *external world* is the world that is composed of representations outside the designer or design agent.

The *interpreted world* is the world that is built up inside the designer or design agent in terms of sensory experiences, percepts and concepts. It is the internal representation of that part of the external world that the designer interacts with.

The *expected world* is the world imagined actions will produce. It is the environment in which the effects of actions are predicted according to current goals and interpretations of the current state of the world.

These three worlds are recursively linked together by three classes of processes. The process of *interpretation* transforms variables which are sensed in the external world into the interpretations of sensory experiences, percepts and concepts that compose the interpreted world. This is done by the interaction of sensation, perception and conception processes (Gero and Fujii 2000). The process of *focussing* focuses on some aspects of the interpreted world, uses them as goals in the expected world and suggests actions, which, if executed in the external world should produce states that reach the goals. The process of *action* is an effect which brings about a change in the external world according to the goals in the expected world.

Situated design computing has the capacity to be the basis of computational models of designing that more closely account for the observed behaviour of designers. It has the capacity to model to changes in interest of the designer as he /she observes emergent structures in their designs, structures that could not have been predicted at the outset. Such emergent structures map onto the concepts of "where you are when you do what you do matters" that is one of the bases of the interaction process in constructive memory..

4. Putting Constructive Memory into Tools

The implications of constructive memory when utilised in a tool are profoundly different from fact or experience memory. The combination of situatedness and constructive memory provides the basis for a new way of thinking about how to build tools and what we can expect from them – tools that adapt their behaviour to their use through interaction rather than the original programmer coding up the possible uses.

With situated computing we can now produce tools that learn from their experiences and apply what was learnt within both like and new situations. Take as an example a building code checker. If it is used initially only on single storey house layouts it will develop experiences applicable to single storey house layouts. These experiences will include how it has been used. These experiences are developed in conjunction with the situation which here includes notions of houses and single storey buildings. The more it is used on single storey house layouts the more its experience and the connection of the situation to the experience is reinforced. So that when it used for another like building, which it can determine by the situation, it will use its reinforced experience that may include which part of the code needs to be checked based on which sorts of rooms are likely to exist in a house. In this sense the tool develops "expertise", ie the ability to perform in a specialized manner as the tool has specialised itself to the task through its interaction with the task.

If the same tool were initially used to check single storey hospital buildings against the code rather than houses it would develop experience related only to hospitals. When this tool was used with houses its performance would be that of a novice even though its performance on hospitals would become that of an expert.

Acknowledgements

This work is supported by a University of Sydney Sesqui Research and Development grant. Discussions with members of the Key Centre of Design Computing and Cognition, including Rob Saunders, Greg Smith, PakSan Liew, Udo Kannengiesser, Rabee Reffat and Mary Lou Maher are gratefully acknowledged.

References

Clancey, W. : 1997, *Situated Cognition,* Cambridge University Press, Cambridge.

Gero, J. S.: 1990, Design prototypes: a knowledge representation schema for design, A*I Magazine*, **11**(4), 26-36.

Gero, J. S.: 1996, Design tools that learn: A possible CAD future, *in* B. Kumar (ed.), *Information Processing in Civil and Structural Design,* Civil-Comp Press, Edinburgh, pp. 17-22.

Gero, J. S. (ed.): 2000, *Artificial Intelligence in Design'00,* Kluwer, Dordrecht.

Gero, J. S. (ed.): 2002, *Artificial Intelligence in Design'02,* Kluwer, Dordrecht.

Gero, J. S. and Fujii, H:.: 2000, A computational framework for concept formation for a situated design agent, *Knowledge-Based Systems,* **13**(6), 361-368.

Gero, J. S. and Kannengiesser, U.: 2002, The situated Function-Behaviour-Structure framework, in Gero, J.S. (ed.), *Artificial Intelligence in Design'02*, Kluwer, Dordrecht, pp 89-104.

Liew, P. and Gero, J. S.:2002, An implementation model of constructive memory for a situated design agent, *in* J.S. Gero and F. Brazier (eds), *Agents in Design 2002*, Key Centre of Design Computing and Cognition, University of Sydney, Australia, pp.257-276.

Maher, M. L,. Smith, G. and Gero, J. S.: 2003, Designing 3D virtual worlds as society of agents, *CAADFutures 2003* (to appear)

Rosenfield, I.: 1988, *The Invention of Memory,* Basic Books, New York.

Saunders, R. and Gero, J. S.: 2001, A curious design agent, *in* J.S. Gero, S. Chase and M. Rosenman (eds), *CAADRIA'01*, Key Centre of Design Computing and Cognition, University of Sydney, pp. 345-350.

Schön, D.A. and Wiggins, G.: 1992, Kind of seeing and their functions in designing, *Design Studies,* **13** (2), 135-156.

This is a copy of the paper: Gero, JS (2003) Situated computing: A new paradigm for design computing, *CAADRIA03* (to appear)