

Design Tools as Situated Agents That Adapt To Their Use

John S Gero

*Key Centre of Design Computing and Cognition
University of Sydney
Australia*

<http://www.arch.usyd.edu.au/~john/>

Abstract. Design tools have been built based on a paradigm that is founded on the notion that the tool is unchanged by its use. Humans build up experience in using tools and adapt their use of a tool based on that experience. The claim is that the tool should also adapt based on its experience of its use. This paper founds the notions of tool adaptation on concepts drawn situated cognition.

Keywords. Design tools; adaptation agents; situated cognition; constructive memory.

Introduction

Computational support for design has a long history, almost as long as modern computing. In the 1950s computers were used during the analysis of roof of the Sydney Opera House. Since then there has been a continuous development of computational support for design that has manifested itself in design tools. There are four classes of design tools: those that deal with generic tasks not uniquely associated with design (word processing for example); those that deal with analysis tasks (for example energy analysis of a proposed design); those that deal with synthesis tasks (for example layout of spaces) and those that deal with documentation tasks (CAD systems for example). All of these tools have one feature in common: they have been built based on a paradigm that is founded on the notion that the tool is unchanged by its use. This paradigm is drawn from the view that the computational embodiment in the tool contains objective knowledge that is independent of its application. For example, the knowledge in a structural analysis package is unrelated to its application or the knowledge in a floor layout program is independent of its application. This has been taken to imply that the application of the tool is independent of its use, ie, it does not matter where it is used since the knowledge stands by itself. In this paper I will introduce another concept that allows the tool to adapt based on its use, ie, the tool will develop “experience” and utilize that experience to guide its further use. Humans certainly build up experience in using tools and adapt their use of a tool based on that experience. The claim is that there are benefits if the tool also adapts based on its experience of its use. The effect of this adaptation is to make the tool both more efficient and more effective. The tool will be more efficient because it will have adapted itself to the class of problems it has previously experienced so that when confronted with a similar problem it will be able to draw on its experience in being used with similar problems. It will be more effective because it will be able to recognize which class of problems it has experience with.

Tools as agents

We will assume that the design tool can be developed as a computational, rational agent, through being wrapped by set of constructs that allow it act as a computational, rational agent. Such a rational agent is a computational system that exhibits autonomy independently of the computational processes it embodies.

Whilst there is no universal definition for the term agent, a commonly accepted view treats agents as intentional systems operating independently and rationally, seeking to achieve goals by interacting with their environment (Wooldridge and Jennings, 1995). The notion of a wrapper allows the agency to be added to the tool so that it does not change the knowledge embedded in the tool.

Situated cognition

In order to achieve the adaptation we seek we draw ideas from situated cognition (Clancey, 1977), a new area in cognitive science. Situated cognition deals with the way humans construct their internal worlds and their interaction with the external world. This is in contradistinction to the notion that the world is “out there waiting to be modeled internally”. Thus, one of the fundamental concepts of situated cognition is the notion that the world within which an individual operates is constructed by that individual and is inside them. We can take these ideas and utilize them as the foundation for a situated agent. Situated cognition uses two concepts that provide the foundation for our agent to adapt to its use. The first is called “situatedness” and the second “constructive memory”.

Situatedness

Situatedness holds that where an agent is and what it does when it is there matters, ie, the agent’s behavior is not independent of where it is, what it does and how it is used. It is dependent both on the world it constructs and on the experiences it has had.

Constructive memory

For example a vision system that recognizes shapes becomes experienced only in those shapes it has been exposed to and not in others. Therefore, when it is exposed next to similar shapes it is experienced in it and will know more readily what to do. When it is exposed to shapes it has not seen before and has no experience with it will take longer to interpret the shapes or may not be able to interpret them at all. Experience in agents is developed through constructive learning. Situatedness involves taking account of both the context the agent is working in as well as its own experience.

In order to understand these ideas we need to change our comprehension of memory. Typically, memory in a computational system is a place filled with things called “memories”. Thus, when a memory is required it is accessed by either knowing its location or its content. Both of these are forms of indexing. In both cases the thing that is the memory is unchanged by accessing it and is unchanged by what it is used for. Even if a memory were accessed one hundred times in a row, the next access would be exactly the same as the first, as far the system was concerned. Constructive memory is a process (rather than an index) that uses the current situation to cue the need for a memory. The memory that results is constructed based on an interpretation of previous experiences through the lens of the situation that required it. The same experiences can be used to construct a multitude of memories. These constructed memories only come into existence when there is a need for them to exist. If there is no need for a memory then it does not exist (Gero, 1999); this is an unusual view of computational memory but accords with ideas derived from cognitive view of memory (Clancey 1997). Constructive memory allows an agent to behave very flexibly in different situations, in ways that fixed memory systems can not. Figure 1 outlines this view of constructive memory.

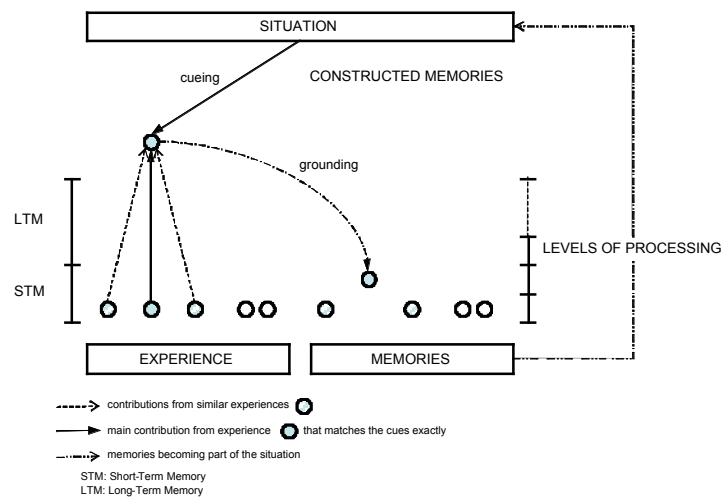


Figure 1. Outline of constructive memory as a process rather than a thing (Liew and Gero, 2002 after Gero, 1999)

Constructive memory is a continuous learning process that reinterprets what has previously been learnt every time new learning occurs. This concept is best exemplified by a quote from Dewey via Clancey (1997):

“Sequences of acts are composed such that subsequent experiences categorize and hence give meaning to what was experienced before”. This memory process forms the foundation of the learning process for a tool that adapts.

Tools that learn

The first step for an adaptive tool is that it learns. As indicated above, constructive memory provides a process that carries out learning in a different manner to many machine learning processes. The learning here is part of the memory of the agent rather than an adjunct to it. A tool that learns and adapts itself has been designed and implemented (Gero, 1996). This tool uses a genetic algorithm that adapts itself to its use through a genetic engineering process that adapts the genetic knowledge in the genetic algorithm. As a consequence, when it is used again on similar problems its behavior improves.

Consider a layout planning problem where a tool that is based on solving it using genetic algorithms is employed. The tool then learns about what makes solutions good and adapts its knowledge to include that knowledge (Gero and Kazakov, 2001). Figure 2 shows how its behavior improves when learning takes place and improves even further when the tool adapts itself based on the knowledge it has learnt. Here the learning mechanism is a novel machine learning approach based on genetic engineering that is a form of unsituated constructive memory. It develops knowledge specific to the experience it was exposed to.

Design tools as situated agents that adapt to their use

The example, the results of which are shown in Figure 2, of how learning can improve the behavior of a tool and how adaptation improves it even further is a demonstration of the utility of the general concept of tools that learn and tools that adapt themselves based on their use. However, what it does not have is any knowledge of the situation within which is being applied.

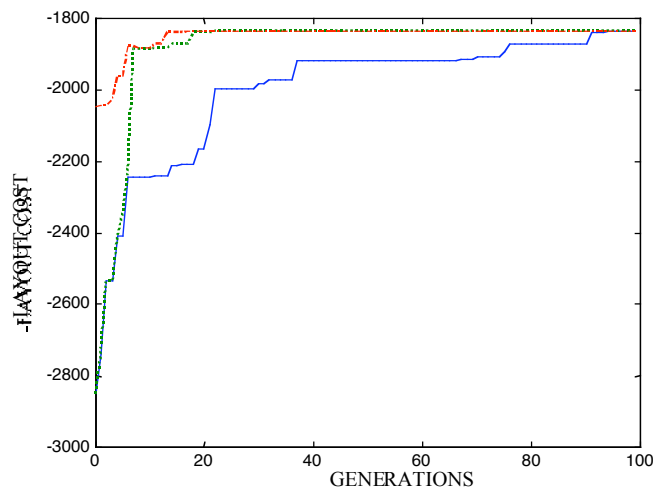


Figure 2: The best layout cost for a spatial layout problem vs generation number for the standard genetic algorithm (solid line), genetic engineering genetic algorithm with learning (dotted line) and genetic engineering genetic algorithm with adaptation (dashed line).

In order to generalize this approach, it is claimed that situated agents that utilize constructive memory are capable of both the learning and adaptation needed to allow tools to continuously adapt themselves to their use. Constructive memory systems have been developed that have behaviors approaching those needed for tool adaptation (Liew and Gero, 2002). Situated approaches have been shown to produce the behaviors expected of them (Gero and Kulinski, 2000). We are now in a position to bring these two concepts together to produce design tools as situated agents that adapt to their use. This would produce a new generation of design tools, tools that change their behavior based on their interaction with their users. This will produce personalized versions of tools.

Acknowledgements

This work is supported by a University of Sydney Sesqui R & D grant.

References

- Clancey, W.: 1997, *Situated Cognition*, Cambridge University Press, Cambridge.
- 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.: 1999, Constructive memory in design thinking, in G. Goldschmidt and W. Porter (eds), *Design Thinking Research Symposium: Design Representation*, MIT, Cambridge, pp. 1.29-35.
- Gero, J. S. and Kazakov, V.: 2001, Design knowledge acquisition and re-use using genetic engineering-based genetic algorithms, in R. Roy (ed.), *Industrial Knowledge Management - A Micro Level Approach*, Springer, London, pp. 107-122.
- Gero, J. S. and Kulinski, J. : 2000, A situated approach to analogy in designing, in B-K. Tang, M. Tan and Y-C. Wong (eds), *CAADRIA2000*, CASA, Singapore, pp. 225-234.
- 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.
- Wooldridge, M. and Jennings, N. R.: 1995, Intelligent agents: Theory and practice, *Knowledge Engineering Review* 10(2), pp. 115-152.

This is a copy of the paper: Gero, JS (2003) Design tools as situated agents that adapt to their use, *in* W Dokonal and U Hirschberg (eds), *eCAADe21*, eCAADe. Graz University of Technology, pp 177-180.