VIRTUAL ENVIRONMENTS USING SITUATED COMPUTING CAN CHANGE WHAT WE DESIGN

John S Gero [1]

(1) Key Centre of Design Computing and Cognition
University of Sydney
Australia
+61 2 9351 2328
E-mail : john@arch.usyd.edu.au

Abstract: This paper presents the foundational concepts of situated computing: first-person interaction, constructive memory and situations. It then describes two classes of situated design that differ from other forms of designing: situated interaction design and situated artefact design.

Key words: situated computing, situated design, interaction design, artefact design.

1- Introduction

Virtual environments are increasingly being used in designing (Maher et al VDS). Virtual environments can be utilised in the following ways:

- as simulations of the physical world
- as partial simulations of the physical world
- as simulations of combinations of physical and non-physical worlds
- as simulations of non-physical worlds, and
- as other worlds.

Using virtual environments as simulations of the physical world is normally carried out by scientists wishing to study various physical phenomena. The simulations by NASA are an example of this use of virtual environments, as are those from medicine. Many of these simulations now have a very high degree of fidelity with the physical world. Using virtual environments as partial simulations of the physical world is where we find computer-aided design. Typical CAD models are only partial simulations of what is being modelled. They rarely include, for example, gravity or tolerances. The scope of these simulations is being expanded to include an increasing range of variables that allows closer representations of the physical world.

More recently there has been interest in developing virtual environment simulations of combinations of both physical and non-physical worlds. The most common examples come from the games world, where, for example, teleporting and weaponry sit outside the physical world. In the design domain the use of avatars within models of the physical world is an example of this burgeoning area.

Virtual environment simulations of non-physical worlds range from simple websites through the creation of worlds that look like the physical world but obey different laws than the physical world to information spaces. Such environments open up new possibilities for interactions that do not need to have counterparts in the physical world.

What appear to be simulations of non-physical worlds can be the world itself rather than a simulation of anything. These other worlds can have any behaviours that a designer gives them.

This spectrum from virtual environment simulations of physical worlds to virtual environments as the world itself demonstrates the power and reach of virtual environments. Such worlds are generally designed using a paradigm founded on the concept that the world is fixed and is unchanged by the activities that occur in relation to it. In the physical world, for example, elements degrade with increasing use. In the non-physical world of search engines, the more a website is accessed the easier it is for others to find. When we include human behaviour in virtual environments we find the same deficit. Human behaviour is not fixed, it varies with experience.

This notion of changing behaviour with experience need not be limited to simulations of humans but can also be designed into non-human elements in a virtual environment. This produces a highly adaptive world that is a consequence of both its existence and its experience. Its experience will be a function of what has happened to it through the interaction it has had with humans or elements within itself or with other environments.

The notion of experience being the foundation of part of behaviour comes from the domain of situated cognition [1].

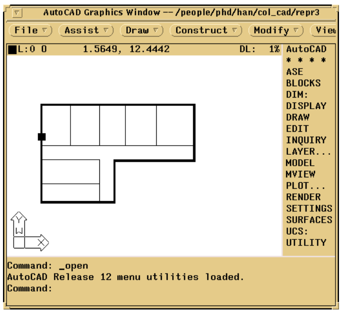
Concepts from situated cognition can form the basis of a new computational paradigm called situated computing that embodies experience. When applied in the design domain it becomes situated design computing.

2- Situated design computing

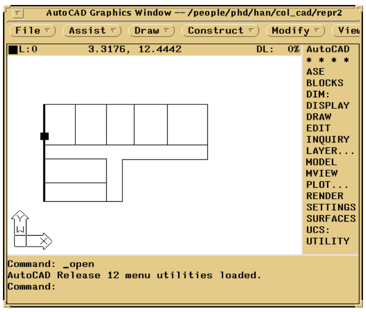
2.1 – First-person interaction versus deductive knowledge

Whilst it is clear that much of human knowledge is objective, in the sense described here, there is a category of everyday knowledge that depends on the person rather than deduction. This knowledge is developed based on first-person interaction with the world. This class of knowledge is sometimes inappropriately encoded as deductive knowledge and when done so often causes a mismatch between the experience of the person who encoded 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 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.



(a)



(b)

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.

The issue here is one of interpretation that has been missing in design computing. A common assumption is that the external 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.

> *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 a design.*

Consider the image in Figure 2. Is it a set of triangular-shaped objects pointing towards to top right with their bases facing downwards to the left? Or is it a set of triangular shaped objects pointing to the top left with their bases pointing to bottom right? Or is it a set of triangular shaped objects pointing downwards with their bases pointing upwards? Each of these interpretations is unique and is not simply a rotation of another interpretation. There are many other interpretations possible.

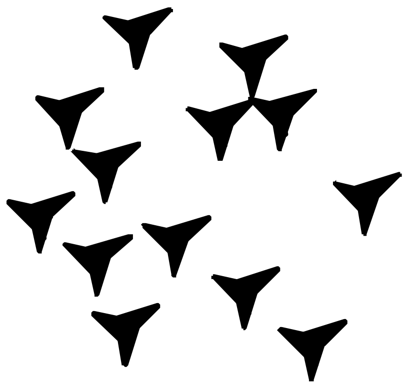


Figure 2: The external world needs to be interpreted before it can be represented

2.2 – Constructive memory

Constructive memory is a major departure from other computational approaches to memory. It turns memory into a continuous learning process. Gero [2] and Liew and Gero [3] outline a model of a memory construction process that carries out the following operations:

- Cueing: The memory system is first cued by a demand from the current situation.
- Activation and selection: A series of experiences is activated and one is selected for use.
- Memory construction: Memory is constructed according to the selected experience.
- Incorporation: The constructed memory is incorporated into the system for subsequent use.
- Grounding: The activated experience is grounded if it is used.

Memory construction commences with a demand for a memory. The memory system is cued according to the current situation. This situation is determined by the current state of the environment and the design agent. The current state of the environment is represented by what information is currently available from the environment. The state of the computation is determined by the knowledge and experiences processed by the memory system. Through cueing, associated design experiences are activated for use during memory construction. One or more of these experiences is selected for subsequent memory construction.

An elementary example of this process in humans is the way a new telephone number becomes easier to recall as it is given out to many people and hence accessed more often. A richer example relates to when a person sees a movie and forms an opinion of it but their opinion changes as they hear other people discussing the movie. Their memory of the movie is no longer based on their own sensate experience but is a construction that relates to their interaction with others.

2.3 – Situations

Designing can be characterised as "a goal-oriented, constrained, decision-making, exploration and learning activity which operates within a context which depends on the designer's perception of the context." [4]. During designing, a designer determines the variables that contribute towards the function, behaviour and the structure of the design according to his/her experiences, knowledge and conception of what is presented to him/her (the situation). These behaviour and structure variables are not chosen a priori but are produced in response to the various situations as they are encountered by the designer. What the designer has done previously, both prior to this design and during the current design process affects how the designer views the situation and what memories he/she brings to bear on the current situation [5].

Much of human design knowledge is individual and based on the first-person interaction with the design as it develops biased by the designer's previous experiences. This is one of the reasons why different designers confronted with the same requirements produce different designs. If designers all produced the same design then we would have catalogues of designs and not require designers to work on each problem separately. The implication of this is that first-person design knowledge cannot be encoded directly into a tool by the tool builder but can only be developed through the experience of doing.

> *The inability of the current design computing paradigm to encompass first-person knowledge has limited the applicability of computers to a restricted class of applications where only objective knowledge plays a role.*

When these systems have been used where first-person knowledge is needed they have failed to produce the efficacy expected because all their knowledge is hard-coded. Another computational paradigm is needed to augment the current objective knowledge-centred one to increase the applicability and effectiveness of computational support for design. The research in this project is targeted at this gap in our technology.

Human designers work within a world that is based on their perceptions. Although much of the world is defined for them in the way it is presented to them, it depends on what the designer thinks is significant as that changes what they think the design issues are and what sorts of means are open to them to design solutions to these problems. This perceived world is an interpretation by the designer of the world external to them biased by their own experiences and knowledge. The designer then produces an expected world as a result of designing. When the expected world produces actions on the external world then a design is realized. This interaction between the designer and the environment strongly determines the course of designing and the resulting worlds and is called the *situation*, whose foundational concepts go back to the work of Dewey [6] and Bartlett [7]. In paraphrasing Clancey [1] we can summarize it as "where you are when you do what you do matters". Figure 3 shows the relationship between these worlds.

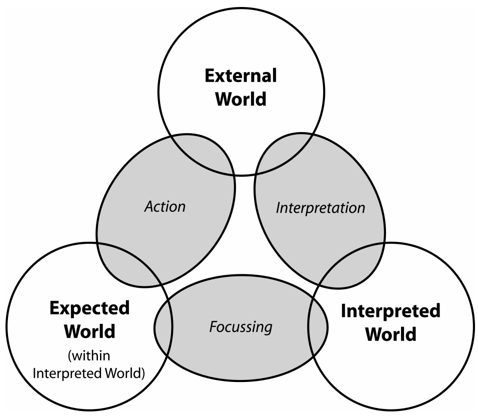


Figure 3: The *situation* arises from the designer's interactions with the external, internal and expected worlds.

These concepts can be used to change how we design and what we design. Two areas of interest are interaction design and artefact design.

3- Situated interaction design

Virtual environments lend themselves to a high level of interaction. Interaction continues to be explored as an area of potential change. There are two classes of interaction of interest here:

- interaction at the interface that changes the interface's response
- interaction with the program changes the program's response.

3.1 – Interaction at the Interface that Changes the Interface's Response

All programs require an interface to allow a user or another program to interact with them. Typically these interfaces are static, ie, they do not change with either the user or their use; they do not adapt. Wizards are an example of interface components that can be created based on the actions of the interface user. These are static responses to user interactions. Recently, various interface systems that learn have been developed and deployed such as the Lumeire project from Microsoft Research [8]. Such systems as these adapt themselves to user interaction through Bayesian learning that aims to capture the uncertain relationships between the goals and needs of a user and observations about program state, sequences of actions over time, and words in a user's query. Interface learning systems have been implemented as agents that behave as personal assistants that cooperate with the user on their tasks rather than merely stand as an interface between the user and the application [9].

Wizards are pre-defined sets of instructions that can be executed without user interaction. Although interface agent-based systems begin to address interactions, these systems over-simplify the learning into building short-cuts for some repetitive and mundane tasks. Despite their efficiencies in some application domains, these systems are not able to assist the design process which is a highly situated activity in which designers often generate new understanding of the design from their interactions with the artifacts they produce. What is needed are interfaces that develop concepts and experiences and are able to use them to produce responses in novel situations. Experiences, here, are perceptual categorizations that can be stably reconstructed, and that can be recomposed in different conceptualizations [1]. In this way the interface can change to met to user's expectations. As the user develops experience so does the interface.

Agents utilising concepts from situated computing offer the opportunity to produce interfaces that develop experience through their use. Interfaces of this kind can be expected to personalise themselves to their individual users. As the user develops expertise so the interface develops experience to match the user's expertise.

3.2 – Interaction with the Program that Changes the Program's Response

Increasingly the response of programs is no longer simply pre-programmed. There is a variety of reasons for this ranging from attempting to make the programs more widely applicable to engaging users more. One common approach it to have the program adapt its behaviour based on the interactions it has had with its users. These adaptations can be based on two classes of behaviours:

- nonlinear responses based on interactions
- situated responses base on experiences derived from interactions.

Nonlinear response is an interesting from of pre-programmed response since the user cannot directly intuit the model of the system's behaviour. A different form of response behaviour can be produced using a situated approach where experience of the interaction is captured, concepts to support that experience are developed and memories constructed. This provides the basis for systems that adapt their behaviour based on their experience rather than in some form of pre-programmed manner. This can be more readily seen in examples where the program interacts with its own environment to develop experience that is used. Figure 4 shows a situated system that changes its response as it experiments with its own environment. In this case it is experimenting with sketching. It commences knowing nothing about sketching but through its interaction with its environment its is able to develop experience that is used to produce increasingly complex sketching behaviour through a constructive process [10], Figure 4.

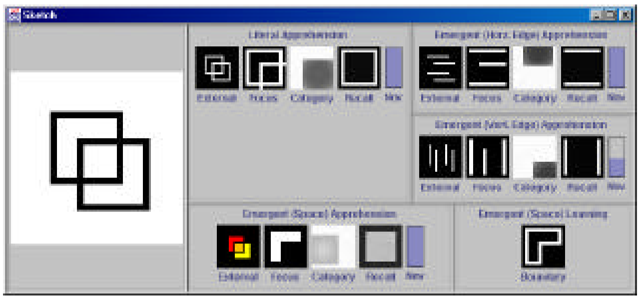


Figure 4: Using experience to construct new shapes. The system is shown only squares but can construct "L" shapes.

The same constructive behaviour can occur as a result of human interaction.

In addition to designing interfaces that have the potential to behave differently we can design situated artefacts similarly.

4- Situated artefact design

Situated artefacts are artefacts that exhibit behaviours that are experientially-based. The design of these involves the same concepts of constructive knowledge and situations that allow artefacts to develop experience and use that experience in new situations. Such artefacts become adaptive. The effect of this is that not all behaviours have to be programmed at the outset, but can develop as a function of what has happened to the artefact up to that point in its existence. The designer now can interact with such an artefact and as a consequence give it experiences that extend its behaviour.

Take as an example a door in a virtual environment. It commences with no knowledge of who to let in apart from the office owner, Figure 5. A the owner lets individuals in, the door learns not only which individuals to let in but also develops concepts as to which class of individuals can be let in and under what circumstances

Figure 5 : A door that adapts its behaviour based on its experience[11]

This opens up a new avenue in the design of virtual artefacts. Artefacts can be designed as agents, where they can function in three modes based on their internal processes: reflexive, reactive, and reflective [12]. Each mode requires increasingly sophisticated reasoning, where reflexive is the simplest.

- *Reflexive mode:* here the agent responds to sense data from the environment with a preprogrammed response – a reflex without any reasoning. In this mode the agent behaves as if it embodies no intelligence. Only preprogrammed inputs can be responded to directly. Actions are a direct consequence of sense data. This mode is equivalent to the kinds of behaviors that are available in current virtual worlds.
- *Reactive mode:* here the agent exhibits the capacity to carry out reasoning that involves both the sense data, the perception processes that manipulate and operate on that sense data and knowledge about processes. In this mode the agent behaves as if it embodies a limited form of intelligence. Such agent behavior manifests itself as reasoning carried out within a fixed set of goals. It allows an agent to change the world to work towards achieving those goals once a change in the world is sensed. Actions are a consequence not only of sense data but also how that data is perceived by the agent. The agent's perception will vary as a consequence of its experience.
- *Reflective mode:* here the agent partially controls its sensors to determine its sense data depending on its current goals and beliefs; it also partially controls its perception processes again depending on its current goals and beliefs; its concepts may change as a consequence of its experiences. The concepts it has form the basis of its capacity to "reflect", ie not simply to react but to hypothesize possible desired external states and propose alternate actions that will achieve those desired states through its effectors. The reflective mode allows an agent to re-orient the direction of interest by using different goals at different times in different situations [13].

5- Conclusion

Situated computing, based on concepts from situated cognition, provides the foundation for an alternate conception of what can be designed in virtual environments. The fundamental change is the introduction of experience through interactions. The interactions can be with humans or with other artefacts. These interactions are the basis of a method of developing new concepts and through the use of constructive memory become part of the artefact's experience. This experience is not static but is re-constructed every time the situation changes. Hence all experiences become part of the artefact's means of responding to situations, whether they be similar or different to other situations.

Artefacts can now respond in unprogrammed ways.

6- References

[1] Clancey W. Situated Cognition, Cambridge University Press, Cambridge, 1997.

[2] Gero, J. S. Constructive memory in design thinking. In G. Goldschmidt and W. Porter (eds), Design Thinking Research Symposium: Design Representation, MIT, Cambridge, pp. I.29-35, 1999.

[3] Liew, P. and Gero, J. S. Constructive memory for situated agents. In AIEDAM, 18(2): 163-198, 2004.

[4] Gero, J. S. Design prototypes: A knowledge representation schema for design. In AI Magazine, 11(4): 26-36, 1990.

[5] Gero, J. S. Towards a model of designing which includes its situatedness. In H. Grabowski, S. Rude and G. Green (eds), Universal Design Theory, Shaker Verlag, Aachen, pp. 47-56, 1998.

[6] Dewey, J. The reflex arc concept in psychology. In Psychological Review 3, 357-370, 1896 reprinted in 1981.

[7] Bartlett, F. C. Remembering: A Study in Experimental and Social Psychology, Cambridge University Press, Cambridge, 1932 reprinted in 1977.

[8] http://research.microsoft.com/~horvitz/lum.htm (last accessed June 2005)

[9] Maes, P. Agents that reduce work and information overload. In Communications of the ACM, 37: 31-40, 1994

[10] Gero, J. S. and Saunders, R. Constructed representations and their functions in computational models of designing. In B-K. Tang, M. Tan and Y-C. Wong (eds), CAADRIA2000, CASA, Singapore, pp. 215-224, 2000.

[11] Maher, M. L., Gero, J. S., Gu, N. and Smith, G. Cognitive agents in 3D virtual worlds. In IJDC http://www.arch.usyd.edu.au/~mary/ijdc03/, 2003.

[12] Maher, M. L. and Gero, J. S. Agent models of virtual worlds, In ACADIA 2002: Thresholds, California State Polytechnic University, Pomona, pp 127-138, 2002.

[13] Gero, J. S. and Kannengiesser, U. The situated Function-Behaviour-Structure framework. In Design Studies 25(4): 373-391, 2004.