Understanding Situated Design Computing and Constructive Memory: Newton, Mach, Einstein and Quantum Mechanics

John S Gero[1]

[1] Key Centre of Design Computing and Cognition, University of Sydney, NSW 2006, Australia
john@arch.usyd.edu,au

Abstract. Situated design computing is an approach to the use of computers in design based on situated cognition. It is founded on two concepts: situatedness and constructive memory. These have the capacity to explain a range of design behaviors but have proven to be difficult to fully comprehend. This paper presents analogies with developments in physics that aim to assist in the comprehension of these foundational ideas. The ideas are drawn from the developments in the notions of space and observations in physics since, to a degree, they parallel the developments in constructive memory and situatedness.

1 Introduction

Design computing is the area of computing that deals with designing and designs. Designs and their representations have been the focus of considerable research that has resulted in a variety of representation models and tools. Designing involves the development and refinement of requirements and approaches, the synthesis of designs as well as the emergence of new concepts from what has already been partially designed. In this designing is not a subset of problem solving – problem solving is a subset of designing. This interaction between partial designs and the design process has been called "reflection" [1]. Reflection is the general term used to describe a designerly behavior that allows a designer to "see" what they have differently to what was intended at the time it was done. Current concepts in design computing make the modeling of this conception of designing very difficult. Other designerly behavior is equally difficult to model using current concepts of design computing. For example: how is that two designers when given the same set of requirements produce quite different designs? Why is that a designer when confronted with the same set of requirements at a later time does not simply reproduce the previous design for those requirements? How is that designers can commence designing before the requirements are fully specified? It is claimed that it is precisely these behaviors that distinguishes designing from problem-solving. Most models of design conflate it with problem solving [2], [3], [4] and are unable to represent the activities that produce these behaviors [5]. As a consequence our computational models of designing and the computational support tools for designing do not adequately match the behavior of designers and are insufficiently effective. This can be seen in the paucity of tools for the early stages of designers where all the critical decisions are taken.

This paucity of tools for the most significant and influential parts of designing is not due to a lack of ingenuity in constructing such tools, it is due to a lack of knowledge about what such tools should embody. This paper claims that the concepts that provide the foundations for situated design computing and hence situated design computing itself provide the knowledge for the development of a new class of tools, tools that have the capacity to support these early stages of designing.

1.1 Situated Design Computing

A new approach is needed. Situated design computing is a new paradigm for design computing that draws its inspiration from situated cognition in cognitive science [6]. It is claimed that it can be used to model designing more successfully than previous approaches. It has the capacity to form the basis of a model that can represent and explain much of designerly behavior. In particular it is claimed it can model [7], [8]:

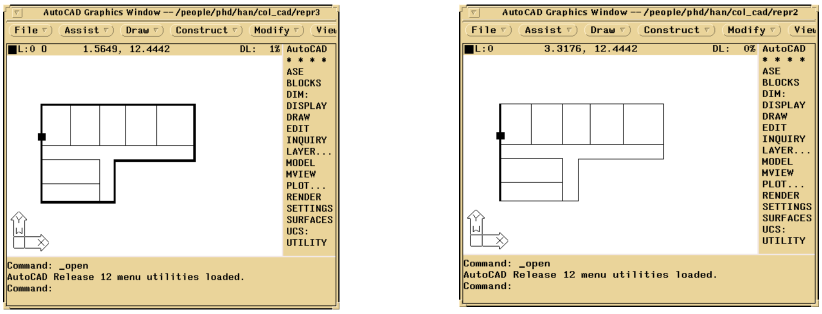
- how a designer can commence designing before all the requirements have been specified
- how two designers presented with the same specifications produce different designs
- how the same designer confronted with the same requirements produces a different design to the previous one, and
- reflection, ie, how a designer can change their design trajectory during the activity of designing.

As a consequence situated design computing warrants further investigation. Situated design computing is founded on three concepts that are new to design computing: acquisition of knowledge through interaction, constructive memory and the situation. The first concept draws its distinction from the source of the knowledge rather than the techniques used in its acquisition. The second introduces a novel notion of memory that conceives of memory primarily as a process rather than the current view of memory as a “thing in a location”. The third introduces the notion of a gestalt view of the designer that influences what the designer “sees”.

1.2 Interaction

Whilst it is clear that much of human knowledge is objective or third-person knowledge of the kind derived in engineering science, 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 [9]. This class of knowledge is sometimes inappropriately encoded as third-person or 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. Fig. 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. Fig. 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 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. Interpretation uses and produces first-person knowledge and is an interaction process.



(a) (b)

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

Interaction is one of the fundamental characteristics of designerly behavior. The interaction can be between the designer and the representation of what is being designed, or it could be between the designer and other designers involved with designing what is being designed, or between the designer and other people, or even between the designer and their own concepts about the design – re-interpretation through reflection.

Interaction is the basis for the development of first-person knowledge, any one of the four kinds of interaction or any combination of them produces first-person knowledge. This first-person knowledge is what distinguishes one designer from another. Tow designers who share exactly the same third-person knowledge can behave markedly differently when presented with the same set of design requirements because of their different first-person knowledge.

1.3 Constructive Memory

Computationally memory has become to mean a thing in a location. The thing can take any form and the location need not be explicitly known. The thing can be accessed by knowing either its location or its content. There are a number of distinguishing characteristics of this form of memory:

- memory is a recall process
- there needs to be an explicit index (either location or content)

- the index is unchanged by its use
- the content is unchanged by its use
- the memory structure is unchanged by its use.

This is in contrast to cognitive models of memory, in particular constructive memory [10], which has the following distinguishing characteristics:

- memory is a reasoning process
- the index need not be explicit, it can be constructed from the query
- the index is changed by its use
- the content is changed by its use
- the memory structure is changed by its use
- memories can be constructed to fulfill the need to have a memory
- memories are a function of the interactions occurring at the time and place of the need to have a memory.

As can be seen constructive memory takes a fundamentally different view of memory than computational memory. Such memories are intimately connected to both the previous memories, called "experiences", the current need for a memory [11] and the current view of the world at the time and place of the need for that memory. Simple examples of some of these characteristics include the following.

Index need not be Explicit, It can be Constructed from the Query. Take the query: find an object with symmetry. There is no need for there to be an index "symmetry" in the memory. The system can use its experiences about symmetry to determine whether it can construct symmetry in objects in the memory. This concept allows for the querying of a memory system with queries it was not designed to answer at the time the original memories were laid down. This is significant in designing as there is evidence that designers change the trajectory of their designs during the process of designing and introduce new intentions based on what they "see" in their partial designs, intentions that were not listed at the outset of process. There are fundamental issues here that are not addressed by fixed index systems. For a novel query to be able to answered it first needs to be interpreted by the memory system using the experiences it has that might bear on the query.

Index Changed by its Use. If the same query is made a multiple times the response to the query should become faster, irrespective of whether it is a constructed index or not; this is a very simple example of how an index is changed by its use. A more profound and useful example of this phenomenon is when a memory is used to construct another, later, memory a new index is created that connects these two memories such that when either is used again the other is associated with it.

Content Changed by its Use. A trivial example is in the case above about symmetry, the index can now become part of the memory. A less trivial example would be in the case where an experience is used in constructing a new memory. The experience is changed by having a link to the newly constructed memory. The experience is no longer the same experience it was before it was used to construct the new memory. That experience can no longer be recalled without its role in the construction of a new memory being part of it.

Memory Structure Changed by its Use. In the example above, not only does the content change but also the structure of the memory. The link between the experience and the new memory changes the structure of the memory system itself such that the way these memories can be used is changed. Later experiences can change what was experienced before. The content of a constructive memory system is non-monotonic.

Memories can be Constructed to Fulfill the Need to Have a Memory. Take the case where a finite element analysis is carried out and both the cost of processing and the result are passed onto the design team leader. The team leader may query whether the analysis was cost effective. There is no memory of this but they can construct a memory in response. Later, if asked whether the analysis was cost effective they can respond directly through a recall-like process. If that query had not been asked earlier there would be no memory to respond to the later query.

Memories are a Function of the Interactions Occurring at the Time and Place of the Need to Have a Memory. Take the example of the cost effectiveness query above. If at the same time as that query being made another member of the design team states that his experience with the analysis group is that they always overestimate their costs, this changes the design team leader's construction of the memory to take account of some discounting of the cost provided.

One way to conceptualize a constructive memory system is as a global, continuously learning, associative system, where all later memories have the potential to include and affect all earlier memories while earlier memories affect later memories. The notion of past memories as fixed entities has to be modified such that all of the past can only be viewed "through the lens of the present", and the present is an encapsulation of the past. This brings us to the notion of a "situation".

1.4 Situatedness

The third concept that provides one of the foundations of situated design computing is situatedness. This is the notion that a designer works within a world of their own making. This world is based on their perception of the world outside and inside them and their behavior is a response to that world. This conceptualized world is called the "situation". Designers do not need to articulate the situation to behave in accordance with it, just as people do not either. Situations can be extrinsic or intrinsic. Extrinsic situations are available for observation, while intrinsic situations are emergent properties of behavior.

A well-known example of a situation is produced when people attempt to solve the nine-dot problem. Consider three rows of three dots equally spaced in both the horizontal and vertical directions. The problem is how to draw lines through all the dots under the following three constraints: using only four straight lines with the pen not leaving the paper. Most people cannot produce a solution. The reason is that they appear to view the world they are in as if the following situation prevailed: no line can pass outside the square (the convex hull) produced by the dots. No one told them this and most people don't even know that they have been working within this situation, but their behavior appears as if it is controlled by it.

Situations are the basis for expectations and interpretations and hence play a dominant role in designerly behavior. Situations can be explicit, ie, known to the designer and affect their behavior, or implicit, ie, not known to the designer and still affect the designer's behavior.

In order to understand situated design computing all three of interaction, constructive memory and situatedness need to be understood. Of the three constructive memory is the most contentious. The next sections draw analogies from the development of the concepts of space in physics that aim to assist in the development of the understanding of constructive memory. It might be argued that the development of the concepts of space in physics in itself is complex and difficult. It is claimed that the base concepts in physics are sufficiently widespread and well understood that they provide a foundation for enhancing the understanding that is trying to be achieved.

2 Understanding Constructive Memory Using Space Analogies

2.1 Newton and Absolute Space: Computational Memory

In 1687 Isaac Newton published *Philosophiae Naturalis Principia Mathematica* [12], in which he introduced the notion of absolute space as the "truest reference for describing motion". Newton's need for absolute space derived from the notion that objects had to have a reference against which some of their properties could be determined. He went on to say: "Absolute space, in its own nature, without reference to anything external, remains always similar and unmovable." This is similar to the view of cognitive realists who claim that there is an objective world that is independent of any observer and is the same for all observers. In Newton's world we could draw a grid and locate all objects on that grid. All object activity relates to that grid. The grid is unchanged by the objects and the grid does not change the objects: it is "always similar and unmovable".

This maps well onto our notion of computational memory, where the location of a thing is memory is absolute and does not depend either on the thing or the "observer" of the thing or whether you access the thing. If we access a thing in memory neither the object nor the memory system is changed, no matter how often we access it. There is no means of accessing it that can change the object. There is no means of accessing it that can change the memory system. It is absolute.

2.2 Mach and No Space: Linked Memory

In 1883 Ernst Mach published *Die Mechanik in ihrer Entwicklung* [13], in which he argued against Newton's absolute space in favour of a completely relative space, where objects are all relative to each other. Mach claimed that objects did not need an absolute space to sit in. They could simply be in relation to each other and that provided the reference that Newton argued was the basis of absolute space. This is equivalent to taking Newton's view as presented with the grid as the reference and removing the grid, leaving only the objects.

This maps well onto the notion of linked or network memory where all things in the memory system are accessed by associations with other things. In linked computational memory things have both absolute locations and links but the absolute locations need not be part of the visible accessing process. Semantic networks [14], [15] are examples of linked memory systems. Accessing a linked memory system does not change its contents, nor does it change anything about the next time you access it.

2.3 Newton and Mach and Gravity: Interacting Memory

In removing the grid Mach did not remove another of Newton's foundational concepts, namely that of interactions between objects: what he called gravitational attraction. In this view objects are linked together by the influence they exert on each other. The introduction of a new object potentially changes the location of some or all existing objects in the system. This influence is bi-directional. Here the newly introduced object's influence is dependent on the individual masses of the existing objects and their relative size in relation to it. A new object will have a small influence if its mass is small in relation to the nearby objects, which will then have a greater influence on it and vice versa. This applies to all the objects in the system.

Currently there is no memory system equivalent to this concept except aspects of constructive memory which we will call interacting memory. A thing in the memory is not only linked to other things in the memory but is influenced by them and influences them.

2.4 Einstein and Gravitational Space: Towards Constructive Memory

In 1916 Albert Einstein published the final version of his *Die Grundlagen der allgemeinen Relativitätstheorie* [16], in which he demonstrated that space was changed by objects (this laid the foundation for our current view of gravity) and that there was an absolute space-time continuum. In one sense this combined and extended the ideas of both Newton and Mach. In its simplest form the theory of relativity re-instated Newton's absolute space but did not keep it absolute. The space itself was now changed by the objects in it. Gravitation was no longer the "attraction of bodies to each other" but a "warping of space" by the masses of the objects.

Continuing with our grid metaphor, the grid was re-instated but was changed by the objects that were still interacting with each other but the interaction was caused by and caused a change in the grid. Imagine a two-dimensional grid with objects hovering over it. The Newtonian view was that the objects interacted with each other but did not change the grid. The Einsteinian view was that the objects changed the grid by warping it into the third dimension creating a three-dimensional grid and the third dimension played a role in gravity.

Consider this gravitational space view:

- a new object affects existing objects
- a new object warps the space around itself

- a new object affects an existing object with an effect that is a function of the size of the new object
- a new object affects an existing object with a relative effect that is a function of the distance to the existing object
- a new object affects an existing object with a relative effect that is a function of the distances and sizes of all the other existing objects
- no object sits by itself unaffected by other objects

Consider a memory system with the following characteristics:

- a new memory affects existing memories
- a new memory affects an existing memory with an effect that is a function of how important the new memory is
- a new memory affects an existing memory with a relative effect that is a function of how important the new memory is in relation to the importance of the existing memory
- a new memory affects an existing memory with a relative effect that is a function of the time distance of the activation of the existing memory to the new memory
- a new memory affects an existing memory with a relative effect that is a function of how important the new memory is in relation to the time distances of the activations and the importances of all the other existing memories
- no memory sits by itself unaffected by other memories.

This brings us close to the characteristics of a constructive memory system. Such a system is more complex than the gravitational space model described above, but one that shares a set of analogically similar characteristics. A constructive memory system has the potential to continuously change as it senses new input that becomes a memory. A constructive memory system has the capacity to meet Dewey's requirement [17] that: "Sequences of acts are composed such that subsequent experiences categorize and hence give meaning to what was experienced before." This is close to the understandings we have about designing. A significant addition to the gravitation space model that constructive memory brings is that new memories can be made entirely out of existing memories, no new sensing is required. This is one example of Schon's reflection.

3 Understanding Situatedness Using a Quantum Physics Analogy

In 1927 Werner Heisenberg published *Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik* [18], one of the cornerstones of quantum mechanics, which displaced Newtonian mechanics, partly by replacing the Newtonian concept of atoms and fields and the implied concept of certainty with an emphasis on subatomic particles and uncertainty. Heisenberg's uncertainty principle states that it is not possible to simultaneously determine both the position and momentum of a particle. In quantum mechanics the location of particles is a function of their viewing. This is an intriguing concept that previously had been associated with social and behavioral science rather than physics.

The concept of situatedness is analogically related to this concept of not having a fixed location and momentum, rather the location or momentum is a function of the observation. A situation is like a particular worldview. A particular worldview affects the interpretation of the memory. The memory could support all manner of worldviews, which one it supports is only apparent when a situation is used to query or access it. Before the memory is queried with that situation it does not have any bias to that situation. The situation of the query biases the memory to produce interpretations that support that situation.

Let us commence with an example drawn from the behavior of a human designer. Take the drawings in Fig. 2. Fig. 2(a) shows the drawing produced by the designer at some point in the design.

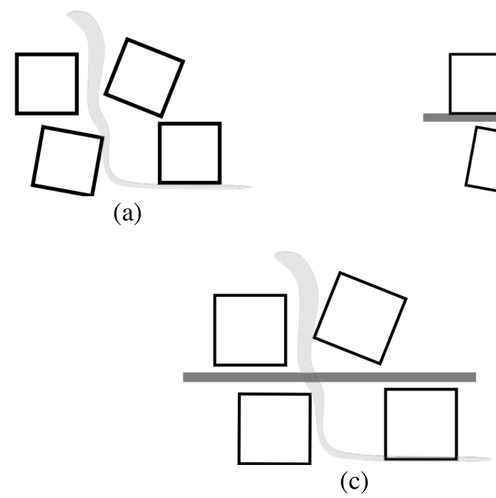


Fig. 2. (a) The original drawing, (b) the designer interprets the drawing as having a horizontal axis; this creates the "axis" situation; (c) the designer now moves the blocks based on the situation of the "axis".

In looking at this drawing the designer reinterpreted what he had drawn not as a series of blocks but as a horizontal axis connecting the blocks, Fig. 2(b). This is a new situation – a new worldview – with this he changes the meanings in his memory of the locations of the blocks and orients them with respect to the axis, Fig. 2(c). Schon and Wiggins [20] observed this behavior on numerous occasions in their studies as did other researchers [21]. This is an example of an extrinsic situation.

Take another example from human behavior. Suppose a designer is shown a picture of pile of rubble that is clearly that of a collapsed house, Fig. 3. The picture is labeled: "result of devastation by Cyclone Larry, 2006". The designer, through their interaction with the picture, is likely to have a response related to the damage caused by nature or similar.

Fig 3. Picture shown to designer.

However, if they were shown the same picture with the label: “example of safety issues in hand demolition of houses” the response would be quite different because the viewer had used a different situation on which to base their interpretation. If the picture was unlabelled it is unclear what the viewer would think without further knowledge about the viewer and the viewer’s experiences.

This behavior is similar to the quantum physics behavior of the location or momentum being a function of the observer, not only of the observed the effect of this concept on memory systems is to change them from static memory to dynamic memory systems.

4 Discussion

Situated design computing is a paradigmatic change in the way we view computation in design. Typically design computing has taken the traditional computational stance of memory being a repository and the primary memory process being that of recall. This has served design computing well, but at the same time has restricted further conceptualization of its development. Novel concepts from cognitive science and in particular situated cognition have opened up new avenues for the development of design computing.

The three foundational concepts of situated design computing are:

- knowledge through interaction
- constructive memory
- situatedness.

Knowledge through interaction moves design computing from encoding third-person knowledge during a computational system’s initial design to including first-person knowledge acquired during interactions continuously through the use and application of the system. The effect of this that such a computational system can adapt its behavior based on its use [22].

Constructive memory provides the foundation for all the activity in situated design computing. It turns memory into a dynamic process that reconfigures itself based on the "experiences" it has encountered. The trajectory of the development of our current understanding of space from physics provides a strong analogy with the development of our understanding of constructive memory [10].

Situatedness provides the foundation for a range of changes in a constructive memory system. It is one of the ways in which reinterpretation of existing memories can take place. In design computing it plays a role in emergence [23], [24], reflection [1] and reinterpretation [5]. Analogies from quantum physics map well onto the base concepts of the situation.

Situated design computing is a different paradigm for computing and the development of computational tools to support designing. It does not displace but rather augments the more traditional design computing paradigm. The distinction between the two is best summarized by treating design computing as embodying third-person knowledge and situated design computing embodies first-person knowledge. Design computing largely uses fixed structures both in the way it is conceptualized and in the way it is implemented. Situated design computing introduces dynamic structures throughout its conceptualization and implementation.

A variety of situated agents have been implemented based on situated design computing. These agents, to varying degrees, embody these concepts and have been applied to:

- designing in virtual environments [25], [26], [27]
- simulation [28]
- modeling expertise [29]
- design optimization [30]
- exploring creativity [31]
- learning [32]
- design ontologies [33], [34] and
- product model interoperability [35].

Projects that are based on situated design computing include:

- interpretation as a design process
- situated design prototypes
- situated design optimization
- design team behaviour.

Acknowledgements

This research is part of a larger project on developing situated design computing and is supported by a grant from the Australian Research Council, Grant No. DP0559885.

References

1. Schon, D. : The Reflective Practitioner: How Professionals Think in Action. Arena, Aldershot. (1995)
2. Pahl, G. and Beitz, G.: Engineering Design : A Systematic Approach (translated). Springer, Berlin (1999)
3. Coyne, R., Radford, A., Rosenman, M., Balachandran, M. and Gero, J. S.: Knowledge-Based Design Systems. Addison-Wesley, Reading (1990)
4. Suh, N.: Axiomatic Design. Oxford University Press, Oxford (2001)
5. Lawson, B.: How Designers Think, 4th edn. Architectural Press, London (2005)
6. Clancey, W.: Situated Cognition. Cambridge University Press (1997)
7. Lawson, B.: Acquiring design expertise. In Gero, J. S. and Maher, M. L. (eds), Computational and Cognitive Models of Creative Design VI. Key Centre of Design Computing and Cognition, University of Sydney, Sydney (2005) 213-229
8. Constructive Memory Group meetings. Key Centre of Design Computing and Cognition, University of Sydney, Australia (unpublished notes) (2006)
9. Wegner, P.: Interactive foundations of computing. Theoretical Computer Science (1998) 192: 315-351
10. Gero, J. S.: Constructive memory in design thinking. In G. Goldschmidt and W. Porter (eds), Design Thinking Research Symposium: Design Representation, MIT, Cambridge, (1999) I.29-35
11. Bartlett, F. C.: Remembering: A Study in Experimental and Social Psychology. Cambridge University Press, Cambridge (1932 reprinted in 1977)
12. Newton, I,: Sir Isaac Newton's Mathematical Principle of Natural Philosophy in His System of the World. trans. A. Motte and F Cajori, University of California Press, Berkeley (1934)
13. Mach, E.: The Science of Mechanics: A Critical and Exposition of its Principles. Open Court, Chicago (1893)
14. Shapiro, S. C.: A net structure for semantic information storage, deduction and retrieval. Proc. IJCAI-71, (1971) 512-523
15. Sowa, J. F.,:(ed) Principles of Semantic Networks: Explorations in the Representation of Knowledge. Morgan Kaufmann Publishers, San Mateo, CA (1991)
16. Einstein, A.: The foundation of the general theory of relativity. In H. A. Lorentz, A. Einstein, H Minkowski, H. Weyl, Principle of Relativity: A Collection of Original Memoirs on the Special and General Theory of Relativity, Dover, New York, (1923, republished in 1952) 109-164
17. Dewey, J.: The reflex arc concept in psychology. Psychological Review. 3 (1896 reprinted in 1981) 357—370
18. Heisenberg, W. : Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik. Zeitschrift für Physik. 33 (1927) 879-893
19. Suwa, M., Gero, J. S. and Purcell, T.: The roles of sketches in early conceptual design processes. Proceedings of Twentieth Annual Meeting of the Cognitive Science Society, Lawrence Erlbaum, Hillsdale, New Jersey (1998) 1043-1048
20. Schon, D. A. and Wiggins, G.: Kinds of seeing and their functions in designing. Design Studies (1992) 13, 135-156.
21. Suwa, M., Gero, J. S. and Purcell, T.: Unexpected discoveries: How designers discover hidden features in sketches. In Gero, J. S. and Tversky, B. (eds), Visual and Spatial Reasoning in Design. Key Centre of Design Computing and Cognition, University of Sydney, Sydney, Australia (1999) 145-162
22. Gero, J. S.: Design tools as situated agents that adapt to their use. In W. Dokonal and U. Hirschberg (eds), eCAADe21, eCAADe. Graz University of Technology (2003) 177-180

23. Holland, J.: Emergence: From Chaos to Order. Perseus Books, Cambridge, MA (1999)
24. Stiny G.: Emergence and continuity in shape grammars. In U Flemming and S Van Wyk (eds). CAAD Futures'93. Elsevier Science (1993) 37-54
25. Maher, M. L. and Gero, J. S.: Agent models of 3D virtual worlds. ACADIA 2002: Thresholds, California State Polytechnic University, Pomona, (2002) 127-138
26. Smith, G., Maher, M. L. and Gero, J. S.: Towards designing in adaptive worlds. Computer-Aided Design and Applications (2004) 1(1-4): 701-708
27. Smith, G.J., Maher, M.L., and Gero, J.S.: Designing 3D virtual worlds as a society of agents. In M-L. Chiu, J-Y. Tsou, T. Kvan, M. Morozumi and T-S. Jeng (eds), Digital Design: Research and Practice, Kluwer (2003) 105-114.
28. Saunders, R. and Gero, J. S.: Situated design simulations using curious agents. AIEDAM (2004) 18 (2): 153-161
29. Gero, J. S. and Kannengiesser, U.: Modelling expertise of temporary design teams. Journal of Design Research (2004) 4(3): http://jdr.tudelft.nl/.
30. Peng, W. and Gero, J. S.: Concept formation in a design optimization tool. DDSS2006 (2006) (to appear)
31. Sosa, R. and Gero, J. S.: Design and change: A model of situated creativity. In Bento, C., Cardosa, A. and Gero J. S. (eds) Approaches to Creativity in Artificial Intelligence and Cognitive Science, IJCAI03, Acapulco (2003) 25-34
32. Gero, J. S. and Reffat, R.: Multiple representations as a platform for situated learning systems in design. Knowledge-Based Systems (20010 14(7): 337-351
33. Gero, J. S. and Kannengiesser, U.: The situated Function-Behaviour-Structure framework. Design Studies (2004) 25(4): 373-391
34. Gero, J. S. and Kannengiesser, U.: A Function-Behaviour-Structure ontology of processes. In J. S. Gero (ed), Design Computing and Cognition'06, Springer (2006) 407-422
35. Kannengiesser, U. and Gero, J. S.: Agent-based interoperability without product model standards. Computer-Aided Civil and Infrastructure Engineering (2006) (to appear)