

1 Of sails and sieves and sticky tape

Bryan Lawson

This chapter concentrates on creative conceptual design and will not deal with downstream issues of detailed technical development or the generation of production information. The title of the Distributed Intelligence in Design symposium used only the word 'design'. It is not until we got into the description of the conference theme that the word 'production' appeared. From then on 'design' and 'production' were as inexorably linked like 'love and marriage', as the song would have it. I challenge that assumption, all the more dangerous because it is implicit rather than explicit. In particular, I am concerned about the dangers of developing knowledge structures and applications for the production stages of construction that then wash back into design.

In a paper very well known in the design research world, Nigel Cross asked us: 'Why isn't using a CAD system a more enjoyable, and perhaps, also more intellectually demanding experience than it has turned out to be?' Nigel argued that CAD may in some cases be quicker, but it is more stressful and there is no evidence that the results are better (Cross 2001).

I have taught in schools of architecture that are privileged to have the most able students of their generation. Whether in Sheffield, in Singapore and China, in Holland and Norway, in Sydney or America, I find the same thing. Students no longer think computers are either difficult or extraordinary; they are just a fact of everyday life. Many architecture students find that computers are not a very appealing part of their design lives. My graduates regularly give voice to a tormenting dilemma. Listing their considerable CAD skills in their CVs often helps them to get a job. But they live in fear of their project leaders discovering this, especially during their years of practical training. They return telling tales of being sat for months in front of a computer exploited as 'CAD monkeys'. They have a plethora of terms for the abuse of computers in design, from 'Photoshop rash' (the over-application of textures and photorealistic skies) through 'Macfontopia' (indiscriminate proliferation of fonts made so easy by the Mac) to 'Modelshop bargains' (an over-reliance on 3-D modelling forms). I have censored the names they have for principal partners who insist on all this nonsense to impress their clients but are unable to do it themselves.

Our students were further discouraged when one of their number won a major national award for his use of CAD and yet, with the same submission, failed his master's degree.

My professional experience is hardly more encouraging. I am part of an international consortium that won the competition to masterplan about 100 hectares of central Dublin known as Grangegorman. The lead architects, Moore Ruble and Yudell, are in Santa Monica; the transportation planners, Arups, and conservation consultants, Shane McCaffrey, in Dublin; the landscape architect, Lutzow 7, in Berlin; the sustainability engineers, Battle McCarthy, in London; and I am in Sheffield. We met as a team roughly once every six weeks, but otherwise relied entirely on IT to communicate across continents and time zones.

What software did we use? Obviously we had an FTP server that held jpegs, Microsoft Office documents and pdfs. The size of such files was already a problem and exchanging CAD files or other active documents was impractical. We largely relied on Word and Acrobat Reader. We used some very basic 3-D modelling software but created inert pdf files for exchange. We often sketched over them by hand, digitised and returned similarly dead files. It worked OK, but relied heavily on the trust established in the face-to-face workshops where we sketched by hand and looked at physical models. How disappointing after all these years!

The vast majority of the software most architects use today is generic. We manipulate pixels and vectors and occasionally use crude solid modellers and generic word processors and spreadsheets. The few big CAD systems are not specifically architectural, although some have what you might call an architectural accent such as the Bentley suites. Even these are really AEC rather than architectural in their way of thinking and working. When we recently did research with architects in the UK using the Bentley suite, we struggled to find any operating the latest version or making sophisticated use of its supposed architectural features.

Design as a cognitive task

From a psychologist's perspective, our view of the possible role of the computer has changed and I want to suggest it is now in need of another paradigm shift to take us forward. The first people in this field (Whitehead and Eldars 1964, Auger 1972) expected that long before now computers would be designing buildings.

More recently, I have worked with cognitive scientists who are in what we might call the 'computation theory of mind' camp. This artificial intelligence theory in essence claims that eventually we will make computers do what our brains can do; the only problem is we have not yet got big enough and powerful enough machines and sufficiently sophisticated software languages. Many of us have felt uncomfortable about this for a while, but each time we threw a new challenge down they would eventually rise to it. 'OK,' we said, 'computers can play noughts and crosses, but they can't play draughts.' They did, so we challenged them to play chess. Of course they did that too. Then we cheated and demanded they beat the best human players. Guess what? They did, although no one seriously claims the software uses human-like cognitive processes.

At last cognitive scientists are seeing design as the challenge that collapses this house of cards. You could trace this argument through Jerry Fodor's *The Language of Thought* (Fodor 1975) and then on to Dreyfus' *What Computers Still Can't Do* (Dreyfus 1992) and Vinod Goel's *Sketches of Thought* (Goel 1995). AI claims that we can represent all useful knowledge through symbol systems and thought through the manipulation of those symbols. Our view now is that it does not seem possible to represent design knowledge and processes in this way. The leap from chess to design is not the same sort of thing as the step from draughts to chess. It is fundamentally different. This is beautifully illustrated through the famous paradox that Bar-Hillel advanced to show the unfeasibility of automatic language translation (Bar-Hillel 1964).

He asks if we could understand the sentence ‘The box was in the pen’. At first it might sound like a transposition error. But if it was in the context of a child looking for a toybox and possibly being in a playpen, then we can work it out. However, there simply is nothing in the symbol collections themselves that gives this away. We have to bring other knowledge into play and the symbols give no clues about that knowledge, what it might be or how it might work. We do not know how to make a computer that could work this out; and yet we find it easy. Designing is full of this sort of knowledge and this sort of thinking. In fact, they are at the very heart of creative designing.

At an RIBA CAAD symposium a software developer prefaced many remarks with the phrase ‘the trouble with architects is...’ I suggested that if the vast majority of architects behaved in the same way there were two possible explanations. The first was that all the most stupid people in the world had by chance chosen to become architects. The second was that perhaps they had adapted to their situation intelligently. So we had better darned well try to understand not just how architects think, but why. This idea offers a small creative leap that may help re-orientate us here. Once we start to think about the cognition of designing rather than of generating production information, we might not see the architect as part of the construction industry but rather as part of the design industry. This is quite a paradigm shift and I think a necessary one.

Lawson and Dorst lay out a description of what constitutes design expertise (Lawson and Dorst 2009). The model we develop shows a series of levels, rising from the novice through the advanced beginner and competent up to the expert and master and, finally, the visionary. One key finding is that designers operating at higher levels of expertise do not simply do the same things as lower-level experts. They are not quicker, better, more accurate or efficient. They actually do quite different things. In a curious way, they think less.

This model fits into a more generic set of ideas about cognitive expertise. De Groot showed that chess grand masters did very little analysis of board situations but rather recognised them (De Groot 1965). Advanced architects similarly recognise design situations. They can see parallels with other situations they know well. That knowledge about situations also incorporates ideas that in chess would be thought of as gambits, or bits of solutions that can be used, each having advantages and disadvantages. Complex situations may be made up of many of these. Architects talk of precedent, by which they mean the panoply of previous situations that can be brought to bear on the case in hand. Unlike lawyers who seek to show the accuracy of precedent, architects seek to interpret it more creatively and to draw it from apparently remote sources. This is a key feature of what we normally describe as creativity.

What this model also shows is that the cognitive support we might need as novices is quite different from that we might need when we are competent and certainly when we are masters or visionary designers. Since I seek excellence rather than the mundane, I am interested in how this affects education and the impact that such ideas might have on the higher levels of architectural design.

What is so different about design?

A key question you might ask here is: ‘What is it then that is so different or special about designing as a cognitive task that makes architects think in such peculiar and infuriating but ultimately fascinating ways?’ The answer to this question is long and complex, but some key points can be developed here with specific reference to how we might develop computer tools to aid distributed intelligence in design.

Design is not like chess. When I was recently designing a garden shelter, I had just spent time in Bali looking at their special way of designing traditional houses and temples. I had seen the

Pondoks crafted by rice workers to allow them shelter from the intense midday sun in the open terraced fields dug out of the lower slopes of the sacred mountain. Knowing this, it should be clear that the design of my 'pondok' was heavily influenced by ideas from Bali, reinterpreted for our landscape and climate and my purpose. There is nothing clever or extraordinary about this; it is the way architects work. Had I been in Africa or South America rather than Asia, it is likely my pondok would have looked different. Design relies, then, on unbounded knowledge. No statement of the problem can symbolically encode information that gives reliable or comprehensive clues as to the kinds of knowledge that might usefully be employed in solving it.

For more problematic features of this world of design cognition, we turn further east to Sydney Opera House. This building is special because it has become so well loved, memorable and symbolic. It represents the unique place in which it belongs, Sydney Harbour, a new culturally progressive Australia, the time it was built and many other ideas. It is fascinating not just as a product but also as a process that has been well documented and teaches us many lessons about designing.

Central to the design are the great concrete sails that simultaneously perform many tasks for Utzon, the architect. They create a magnificent composition sitting perfectly on Bennelong Peninsula jutting out into the very heart of Sydney Harbour. They act as a perfect counterfoil to the famous bridge against which they are so often photographed for that reason. They subtly reflect the sails of the myriad small yachts that often surround the building. Of course, they also house the great spaces of the opera auditorium, the concert hall, the smaller restaurant and the public domain. They create opportunities for solving the tricky problems of threading services through such a complex and demanding set of volumes. They offer a structural system that is self-explanatory, efficient and beautiful when exposed. I could go on.

How can one mind arrive at a single device that simultaneously does so much on so many levels? In truth, the sails perform far better at some of their tasks than others. They leave spaces that have poor acoustics, though that is not really Utzon's fault. They insult and discriminate against the disabled. They make life hell for stagehands; ridiculously, the public approach is from the stage end of the opera house. It is well known that Utzon designed the sails before he knew how to build or even draw them and this was one of the factors that would drive the initial contractor to financial ruin. Again, I could go on.

And yet we forgive the building all these inadequacies because it is so magnificent in so many other ways. To have become one of the best-known buildings in the world with all these faults shows just what a fantastic achievement it is. It narrates a very human story of genius that succeeded in the face of so many difficulties and yet also failed our unreasonable expectations of perfection.

So what do we learn here? Design depends on integrated responses to many disparate factors in one single device in ways that could not possibly be predicted from any symbolic representation of requirements. These factors cannot be measured against criteria with any common metric for success. Which of us can say how many more stairs we are prepared to walk up in order to get the memorable view that Utzon creates for the interval promenaders out in the middle of the harbour?

New ways of communicating with computers

Architects must be using extraordinary mental gymnastics when designing. This implies the existence of a multidimensional cognitive structure that enables multiple ideas to be considered and developed. So if computers are going to assist us in designing, surely we need to converse with them in ways that are at least as sophisticated as we might use when working with other designers. Is this realistic?

Some 30 years ago, my research group developed a suite of CAD programs for designing architecture known as GABLE (Lawson and Roberts 1991). They were founded on the principles of intelligent building modelling and on some key ideas about the nature of architectural design processes. They allowed architects to describe buildings in a variety of cognitive modes observed to be in common usage (Lawson and Riley 1982). Thus one could draw elements such as walls, windows and doors and GABLE would infer a spatial model. Alternatively, one could move, combine or divide spaces and GABLE would update the elemental model.

Back in the 1980s this system was in international use in both practice and education. We learned a huge amount from its use, not so much about CAD but about designing itself and about the complexity of knowledge representation in design. Eventually GABLE failed for a number of reasons, but the main intellectual failure turned on some unwarranted assumptions that are still going unquestioned today.

We must decide the extent to which we expect such systems to be central or peripheral to the creative design process. John Lansdown asked this question decades ago, but few have explicitly attempted convincing answers (Lansdown 1969). He pointed out that there were two fundamentally different strategies we could employ, which he called 'ad-hoc' and 'integrated'. He foresaw a wide range of applications, for example thermal evaluation, daylighting studies, visual form, costing, structure and so on. He realised that such applications need different though overlapping sets of data about features of the design.

Assuming that as designers we might like to be able to see how well our design is working on a number of criteria, how do we input the necessary information? In what Lansdown call the 'ad-hoc' strategy, we input the information needed as we use each individual application. So if we want to perform a simple steady-state thermal evaluation, we would need u-values and the areas of the external skin components. If we want a natural lighting study, then geometry, transmission, reflection data and orientations would be required and so on. This means a very halting process for the designer. Every time you want to examine the design along some dimension, you have to stop and input data.

An illustration of how impossible this would be can be seen from simply observing students. They are struggling to develop an integrated response, a task already almost too demanding for their early level of expertise. They need advice from a range of tutors, about architectural form, construction, structures, environmental control and sustainability. At Sheffield we now have this in-house, but previously structures was a service taught by our civil engineering department. To get advice on structures students had to phone up, make an appointment and then go to the other side of the campus. They did not do it, of course, and we saw many projects that were innovative climatologically but very few that were innovative and creative in structural terms. This ad-hoc idea simply does not work for designers. It would not work for the sails of Sydney Opera House. Utzon could never have used it.

So we turn to Lansdown's 'integrated' strategy and link all the evaluation packages to a single database, now variously described as building information models or n-D models. Each evaluation then runs immediately.

Salford and Sheffield universities collaborated on a research project to create support systems to record and make explicit design rationale (Cerulli *et al.* 2001). There were several objectives. First, the need in a multiprofessional, and often not co-located, team to know who is making what decisions and why. This becomes especially important when things happen in parallel. The classic example is the scenario in which the architect issues a general arrangement floor plan and the M and E engineer starts to run services through routes that the structural engineer is busy blocking. The architect is often left trying to spot this and we all think that CAD clash checking would be the answer.

However, things often get even messier. Our drawings show the decisions but not the reasoning. Later on, someone who may not understand the reasons changes things without realising the damage they are inflicting on the design. Being able to see the thinking behind the design at every stage is far more important than just the clash checking. In the design of Sydney Opera House they built a huge perspex model so everyone could see how spaces were interconnected and related. Today we would think of doing this on a computer. Somehow the physical model still does the job better. Incidentally, this model is now seen as a security risk, since the knowledge it imparts could greatly facilitate terrorism.

The Sheffield–Salford project worked with the Bentley software and logged the complete state of the model, traced all additions and changes and recorded the rationale. Instinctively, we decided to plug our software into the CAD software. All very logical: as you called a routine to add, delete or edit a model element, you automatically accessed the rationale capture software. However, field trials revealed that this was hopeless. More often than not, key decisions were taken away from the computer model.

By way of illustration, you can see a most creative process at work in the Philadelphian offices of Bob Venturi when designing his famous extension to the National Gallery in Trafalgar Square (Lawson 1994). One of Venturi's key forming concepts in this design is how the new architecture relates to the existing Wilkins' building, so famously described by Prince Charles as 'a much loved friend'; this, of course, when he so unfairly criticised the previous competition-winning design by ABK. Venturi had the original façade computer modelled, plotted out and cut up into pieces with scissors. These pieces were then stuck around his new physical models with sticky tape. How ironic to see a computer metaphor being used far more creatively in its physical reality.

The normal situation, then, is that key design decisions are often made over sketches or physical models, on telephone calls or at meetings, on drawing boards and sketchpads. Mostly they are the results of 'conversations' involving much talking and waving hands in the air. Often someone is deputed to input the new state of affairs into the computer model. At that time the rationale is either not available or it is incorrectly guessed. The computer model is simply not where the action is.

A further problem with the integrated model is that it quickly becomes a production tool rather than a design aid. Designers find that you have to specify not only geometry but also materials and construction in such depth that you have effectively generated not a design model but a production model. Does this really matter? Well, yes it does. It is simply unfeasible to argue that tools do not have an impact on processes and that processes do not have an impact on the end product, the architecture itself.

Take the representation of free-form design. Surely this must promote creativity? At last, sophisticated mathematics allows us to compute locations on irregular surfaces. Gehry Technologies have contributed to the migration of ideas from aeronautical design into architecture. Bill Mitchell claims that the use of this in Frank Gehry's 'remarkable late projects will ultimately be remembered not only for the spatial qualities and cultural resonances they have achieved, but also for the way in which they have suggested that everyday architectural practice can be liberated from its increasingly sclerotic conventions' (Mitchell 2001).

Of course he is right, Frank has created a remarkable new form of architecture. But look more carefully and there are two problems here. First, Gehry does not himself use this software in design. Indeed, Zeara claims that 'the computer was introduced into Frank Gehry's office in a way that would not interfere with a design process that had been evolving over thirty years' (Zeara 1995). Lindsey tells us that 'Gehry does not like the way objects look in the computer' and

that he avoids looking at the computer screens in the office (Lindsey 2001). He not as eccentric as you might think. My work on Santiago Calatrava, one of the most creative minds in free-form design, shows that he also does not like to use computers. This man is even a fully qualified engineer as well as being an architect, but he prefers to sculpt physical models and only relies on computers for finite element analysis.

This is quite understandable, since software driven by the complex mathematics of such esoteric devices as Bezier curves or non-uniform rational B-splines is hardly user-friendly. The input of control point locations and tensions on curved patches is not for the fainthearted! It is certainly not intuitive and far from the 'conversation with the drawing' discussed earlier. So Gehry designs with much more plastic materials such as crumpled paper and other expert users have to negotiate these into the computer. Calatrava collaborates with a Swiss watch-maker turned modeller who has become an integral part of his creative process.

The second problem here is in Mitchell's reference to the sclerotic conventions of architecture. He wants us to believe that all architects, their clients and users are desperate to get away from geometry in architecture. There is no evidence to support such a thesis. Instead, I will present one rather powerful anecdote in contradiction. Having become very interested in what Frank Gehry and Jim Glymph are doing, I realised that a very famous piece of architecture indeed could now have been realised very differently. In his original submission for Sydney Opera House, Jorn Utzon included a model showing irregularly curved surfaces for the sails. It is well documented that once the competition was won, Utzon and Ove Arup puzzled over how to build these (Weston 2002). Eventually they resorted to mapping them all onto the surface of a single sphere. Not long before he died, I showed Utzon that the Gehry technology software would now allow him to build the original design and I asked what choice he would make today. The answer was emphatic. He would keep the geometry and the design as finally realised. He was always looking for some rationality and order. To this day, of course, the stunning result is as he wanted, a combination of romantic reference and yet coherent order. As Utzon said in his response to me: 'I am *not* Frank Gehry.' Thank goodness we have been lucky enough to have both of them, but clearly Mitchell's implied idea that we all want to become just like Gehry is simply another example of technology push rather than market pull.

Unfortunately, that technology push can then extend through into the architecture itself. Take the case of the Opera House in Singapore. The design of this can be summed up as a couple of auditoria in boxes standing inside a huge glazed upturned kitchen sieve. When working at the National University in Singapore, I attended a long lecture from one of the software engineers employed on this project. It was their software that had led the architect to appreciate the feasibility of this design in this context. Remember that Singapore has a hot-wet tropical climate with temperatures around the year in the 30s, very high humidity and often daily torrential rain. A glass dome is hardly the first form that comes to mind in such a climate. Compare such an idea with the environmentally sustainable work of Ken Yeang, who tries to create a new regional identity for south-east Asia (Yeang 2006). In a kitchen sieve every cell is unique, as the square grid is resolved onto a curved surface. So in the Singapore dome every cell must have its own tailor-made shading device shaped to its cell size, and of course orientation in order to avoid unbelievably high solar gain. The software to achieve all this is indeed very clever!

The lecture reminded me of much of the silly technology push that surrounds us. The plethora of applications for the iPhone is a wonderful demonstration. I have an app that gives the phone the wonderful capability of acting as a spirit level. 'Very clever,' says everyone who looks at it. Of course, I have never used it. We still treat computers and their smaller palm-top offspring like circus animals trained to perform apparently clever but pointless tasks.

The nature of the computer model itself brings yet further potential remoteness from real design decision making. The art critic Adrian Stokes introduced the delightful distinction between what he called ‘modellers’ and ‘carvers’ (Stokes 1934). These are two distinct forms of thinking about space and form. Modellers would assemble a building from its components; carvers would craft it from its materials. The sculptor, who carves works with the grain of stone or wood, understands the material and even feels the way it wants to be. The great American architect Louis Kahn told us to ‘let a brick be what a brick wants to be’. In other words, he called for architecture that was carved, that worked with the grain of its materiality.

Conversations with the situation

We have a further pervasive problem: a fundamental misunderstanding about the nature of design expertise. In the mid-twentieth century when we first started to explore the nature of design processes, the dominant paradigm was that of problem-solving. Those of us who continue to research this field no longer see this as the only, or even the dominant, way of explaining the creative processes used by designers in general and architects in particular. Sadly, the developers of CAD have not caught up. Implicit in so much of the software is a problem-solving view. The newer view of reflective practice, pioneered and championed by Donald Schön, leads us in quite different directions (Schön 1984). In this paradigm the architect is not so much solving given problems as discovering them in parallel with developing solutions and even through the creation of solutions. In fact, already the words problem and solution are uncomfortable in this world. We now prefer to talk of ‘situations’ in which needs and possible actions are seen as existing in creative tension. Schön talked of reflective practitioners having ‘conversations with the situation’. In the case of architects this situation is often assumed to be represented through drawings, though other media are used too.

If we are going to support this process, we need to understand the methods of knowledge representation used in the drawings architects generate, not for others to see, but to develop their conversation.

Goel has compared the way designers work using ordinary manual sketches with the way they work using very simple computer drafting programs of the vectoring type, namely in this case MacDraw (Goel 1995). Six graphic designers were set the task of designing tourist posters while six industrial designers were asked to design a desk clock and a toddler’s toy. Goel analysed all the drawings produced by both groups of subjects using both manual and computer-based drawing systems. He showed that the drawings done with MacDraw were less dense and ambiguous than those completed by hand. This will not surprise anyone with any skill in drawing who has tried to use such software.

Disturbingly, Goel also showed that this had an impact on the nature of the design thinking and was in turn likely to affect the eventual outcome. The designers using MacDraw made significantly fewer ‘lateral transformations’ than their manual sketching counterparts. That is to say, they tended to persist with an idea for longer, ‘vertically transforming’ it. The inference here is that the less ambiguous MacDraw system allowed the designers less opportunity to ‘see’ different interpretations of their drawings. As a result, fewer ideas were explored in the process in roughly the same period of time.

Bilda and Demirkan tested designers on an interior design task using both manual drawing and a vectoring-based CAD system known as Design Apprentice (Bilda and Demirkan 2002). A retrospective reporting technique was used to get subjects to recall and describe their intentions

by watching a video of the protocol. This study again showed fewer 'cognitive actions' when using the digital media.

A reasonable conclusion is that existing CAD systems use symbolic representations that do not map well onto the mental symbolic representations used by designers. As a result, working with such systems leads to a less rich mental world, since the drawings 'talk back' to us in less suggestive ways. Put simply, the conversation with the computer was less rich than the conversation with a piece of paper.

Kvan demonstrated something that raises other profoundly disturbing questions for the CAD movement (Kvan *et al.* 2003). Architecture students worked in groups using computer-based communications. One group had only text-based technology, while the other could exchange graphical information too. The results of the text-based groups were consistently more creative, original and interesting than those from the groups exchanging graphical information.

A more positive approach

So far my analysis suggests that attempting some computer aid that sits right at the centre of the creative design process is deeply problematic. This is perhaps a rather harsh lesson. First we had to give up the idea of computers designing. Now I am asking you to give up the idea of computers even helping us to design, at least in some central role.

So how can we be a bit more positive? The design expertise model suggests that one of the key tasks at the early stages of expertise development is the acquisition of precedent and of schemata that are used to organise them cognitively. In simple terms, students of architecture need to see and study a wide range of buildings, places, designed and natural objects and other cultural artifacts. While a few geniuses may manage without this, most of us cannot. A student who has not studied the Sydney Opera House, for example, is unlikely to be able to appreciate the ideas that it taught us. These would certainly include the notion of rationalising curved surfaces, how to compose them and the very powerful idea of a free-floating form hovering above a solid plinth. Again, I could go on.

Traditionally students learned these things through extensive travel, scouring magazines and journals and so on. They were always encouraged to carry with them a sketchbook. No architect I admire does not carry such a thing for the immediate and quick sketch to record and analyse. Excellent examples can be found from John Outram looking at Corb's Ronchamp and Bucky Fuller's Dymaxion House (Lawson 1994).

The digital camera, its connection to the computer and all the easy image-manipulation software linked to internet searches make for a different world. This technical advance also poses huge dangers, however. The student who had to travel saw in the flesh, as it were, a totally different experience to a Google Images file. The sketch relied on an eye-brain-hand process that forced a degree of analysis and brought understanding. We now have a generation of students arriving at schools of architecture who have not learned to sketch, have not learned to see, have not learned to analyse, in fact have not learned.

All is not lost. We have used simple solid modellers in exercises with novice students in which they build models of buildings and then deconstruct them to explore the conceptual structures that underpin their design ideas. In some ways this is very successful and illustrates an important principle of using computers that I have always thought desirable: we can actually do something new. Unfortunately, it is also still very crude. The software is far better at some kinds of geometry than others. It works brilliantly in this example with the architecture of the De Stijl movement.

We need to develop software that is specifically targeted for this purpose, not for designing as such but for recording and analysing architectural form.

All this begs another hugely important question about the way we search for information. I used many of the same fundamental concerns to analyse how architects use libraries (Lawson 2002a). There are important lessons that many university librarians find as difficult to accept as some CAD enthusiasts do. My university built a major new library recently, except that we are not allowed to call it that. It is an 'information commons'. The university then wanted to split our collections between the old and new in terms of undergraduate and postgraduate. When I refused to agree to give the librarian the list of undergraduate texts for architecture, he thought I was just being awkward. Not so. First, there is no overall undergraduate subject textbook for architecture and there never will be. Secondly, our students look at the latest journals. You would not expect either of these things in an undergraduate psychology course. Our students take few books out but spend hours in the library scanning, browsing and then sketching and arguing. The library is actually an integral part of the design studio.

Let us see at another example higher up on the expertise development staircase. When studying the work of one of our most successful and creative architects, I visited the office of his practice and heard three different people use the word 'belvedere'. There is nothing extraordinary about this word, but even in an architect's office this seemed too much of a coincidence. It was an example of something we have come to realise more clearly recently: expertise in design is not held only inside single heads but collectively and socially in organisations. This word brilliantly and incredibly efficiently stands for a whole set of architectural ideas that clearly this practice had been talking about. If you were not in on this you simply could not contribute to the design process they were using. This is real distributed intelligence at work in design.

So how do designers search for information and how might we help them? It is, of course, a frustrating process. Even though we now have Google Images, we have search it through words in the titles of the images. How we understand and express our desire for information, use this in conversation and searching and then send ideas to other members of the design team turns out to be a key obstacle. This is hugely challenging and in my view a far more important and interesting problem than most work done currently in CAAD.

At a higher level of expertise we see another problem: the assumption that all knowledge about design is encapsulated in drawings. The wonderful Czech architect Eva Jiricna is widely recognised for her original and creative output (Lawson 1994). I select her as an example because Eva is no Luddite. She is trained as an engineer and is well known for her high-tech interiors for fashion guru Joseph and others. She has described to me how she uses words more often than pictures to discuss ideas with clients until the design is fairly well developed: 'I try to express in words what they want, and then I try to twist it into a different statement and then draw it.' However, this is not some eccentric exception. The very successful product designer Richard Seymour told how he and his partner entered the competition to design the front end of the high-speed train. They astonished the client by not using any images at all at the interview. They merely said that their train 'would make schoolboys want to be engine drivers again and that it would be heroic in the manner of Concorde'. They won the job!

An interest in how we manipulate knowledge in design through words rather than pictures led us to investigate a computer program that could hold a useful and creative conversation with a designer (Lawson and Loke 1997). We wanted to know what building blocks and methods of knowledge representation would be needed. During one research seminar a colleague who is responsible for teaching CAD declared the whole thing a waste of time. 'Go into any architect's office,' he said, 'and 90 per cent of what you see will be drawings.' 'Yes,' I replied, 'but 99 per cent

of what you hear will be words.' The problem here is that the words do not get recorded, and the drawings are left behind to claim all the glory. We have become so obsessed by our cleverness at training the circus animal computer to draw that we have missed the point completely about how designers actually think and work and what tools they might want.

From our model of expertise, we can see that software designed to help at one level of expertise may not necessarily be so useful at another level. A generic problem might be that much of the software ostensibly created for architects to design with is actually written by people who are hardly masters of design, let alone visionary architects. Implicit in the resulting software are many examples of assumptions that the users will work in ways far more likely to be seen at novice rather than expert levels of design. No wonder so many great architects choose not to use CAD.

Our model of design expertise teaches us that different construction industry professionals operate in increasingly disparate ways as they gain expertise. Architects' design knowledge is largely about solutions rather than problems. It is what cognitively we call experiential rather than semantic. We know that these two kinds of knowledge are held differently in the brain, have quite different characteristics and are accessed differently. One sad but dramatic demonstration of this is the way people suffering from severe dementia may be quite unable to remember recent events but can still parse a sentence grammatically and perform accurate arithmetic. Designers, it seems, have a far greater reliance on episodic knowledge than on semantic or theoretical knowledge. Architects simply do not have theories that enable them to get from problem to solution. There really is a difference between the thinking of most engineers and most architects. Architects will commonly tell you that an engineer who thinks architecturally is an incredibly valuable asset in the design team. People like Tony Hunt and the tragically departed Peter Rice are often cited as rare examples of such ability. Software that is somehow capable of bridging these gaps might be an exciting possibility. However, here I want to go one step further.

Zeisel perceptively showed us that the most important and critical gap is not between specialist consultants but between us, our paying clients and users (Zeisel 1984). For those building typologies where the complexity of use is both high and critical, this becomes one of the most serious obstacles to really good design. I have spent a good deal of my time working on this problem in the twin areas of healthcare and education. Hospitals have been heavily researched in recent years (Lawson 2002b). We now maintain a research database for the Department of Health and it has in the region of 1000 pieces of work that in some way link architectural design issues to health outcomes, patient satisfaction levels, quality-of-life issues, staff effectiveness and job satisfaction. Others have shown that if we take all this research together and apply it in the design of new hospitals, we should be able to save something in the region of 20–25 per cent of operating costs with an increase at less than 5 per cent of capital costs (Berry *et al.* 2004). For such buildings characteristically the capital cost is exceeded by the operating costs in less than two years (Lawson 2004).

Different kinds of knowledge about all these issues exist in the heads of people with many roles. We have devised a series of web-based tools now used by the Department of Health for the briefing, design and evaluation of such buildings (Lawson 2007). One of these, IDEAs, serves a useful example here.

IDEAs breaks away from the conventional Department of Health structure of Health Building Notes, Room Data Sheets and all those other production-based tools that focus on compliance with minimum standards rather than promoting excellence, that suppress rather than facilitate creativity. Instead of the multiplicity of named rooms, we broke the whole problem down into fewer than a dozen major activities that turn out to account for almost all areas in healthcare

buildings. These include quite specific places such as ‘consulting/examining/treating’ as well as more generic ones such as ‘arriving’ and ‘circulating’. IDEAs has two major panes. The top pane deals with the challenges and opportunities. The challenges show all the research mentioned above as ways of understanding what people are trying to do in such places. The opportunities discuss the features of the designed environment that can be used to affect this. The bottom pane shows bits of designs that other people have already built that seem to work quite well in some detailed way. So what we do here is to bring problem and solution together in such a way that both client and architect can meet on some territory where they both feel comfortable.

Only the interactive and highly graphical capabilities of web-based applications can offer such an environment. Computers really can help designers. However, it is worth noting one further connection back to our model of design expertise. The model shows that expertise can exist in four different ways. We call this Project, Process, Practice and Profession. What we are beginning to realise is that our CAAD efforts might be misplaced not only because we have tried to embed them too much, but because they are too focused on the ‘project’. The positive examples I have been giving here suggest that we might help more by concentrating on ‘process’ and, even more interestingly, on ‘practice’, realising that students need to develop this differently to competent and expert designers. IDEAs even shows how we can develop a further level of ‘professional’ expertise that exists beyond a single project, outside any one single practice, and yet it can be used to bring all that distributed intelligence through a new process to individual projects.

References

- Auger, B. (1972). *The Architect and the Computer*. London: Pall Mall.
- Bar-Hillel, Y. (1964). *Language and Information*. Reading, MA: Addison Wesley.
- Berry, L. L., Parker, D., Coile, R. C., Jr., Hamilton, D. K., O'Neill, D. D. and Sadler, B. L. (2004). ‘The business case for better buildings’. *Frontiers of Health Service Management* **21**(1): 1–24.
- Bilda, Z. and Demirkan, H. (2002). ‘An insight on designers’ sketching activities in traditional versus digital media’. *Design Studies* **24**: 27–50.
- Cerulli, C., Peng, C. and Lawson, B. (2001). ‘Capturing histories of design processes for collaborative building design development: Field trial of the ADS prototype’. In B. de Vries, J. van Leeuwen and H. Achten (eds) *Computer Aided Architectural Design Futures 2001*. Dordrecht: Kluwer Academic Publishers, pp. 427–38.
- Cross, N. (2001). ‘Can a machine design?’ *MIT Design Issues* **17**(4): 44–50.
- De Groot, A. D. (1965). *Thought and Choice in Chess*. The Hague: Mouton.
- Dreyfus, H. L. (1992). *What Computers Still Can't Do: A Critique of Artificial Reason*. Cambridge, MA: MIT Press.
- Fodor, J. A. (1975). *The Language of Thought*. Cambridge, MA: Harvard University Press.
- Goel, V. (1995). *Sketches of Thought*. Cambridge, MA: MIT Press.
- Kvan, T., Wong, J. T. H. and Vera, A. H. (2003). ‘The contribution of structural activities to successful design’. *International Journal of Computer Applications in Technology* **16**(2/3): 122–6.
- Lansdown, J. (1969). ‘Computer-aided building design: The next steps’. *RIBA Journal* **76**(4): 138–40.
- Lawson, B. R. (1994). *Design in Mind*. Oxford, Butterworth Architecture.
- Lawson, B. R. (2002a). ‘Architecture libraries from a university perspective’. *ARCLIB Bulletin* **11**: 1–4.
- Lawson, B. R. (2002b). ‘Healing architecture’. *The Architectural Review* **CCXI**(1261): 72–5.
- Lawson, B. R. (2004). ‘Assessing benefits in the health sector’. In S. Macmillan (ed.) *Designing Better Buildings*. London: Spon Press, pp. 100–6.

- Lawson, B. R. (2007). 'Design indicators'. In D. Stark (ed.) *UK Healthcare Design Review*. Glasgow: Keppie Design, pp. 88–95.
- Lawson, B. R. and Dorst, C. H. (2009). *Design Expertise*. Oxford: Architectural Press.
- Lawson, B. R. and Loke, S. M. (1997). 'Computers, words and pictures'. *Design Studies* 18(2): 171–84.
- Lawson, B. R. and Riley, J. P. (1982). 'ISAAC: A technique for the automatic interpretation of spaces from drawn floor plans'. *CAD82 Conference Proceedings*, Brighton: IPC Press.
- Lawson, B. R. and Roberts, S. (1991). 'Modes and features: The organization of data in CAD supporting the early phases of design'. *Design Studies* 12(2): 102–8.
- Lindsey, B. (2001). *Digital Gehry: Material Resistance/Digital Construction*. Basel: Birkhauser.
- Mitchell, W. J. (2001). 'Roll over Euclid: How Frank Gehry designs and builds'. In J. F. Ragheb (ed.) *Frank Gehry, Architect*. New York: Guggenheim Museum Publications, pp. 352–63.
- Schön, D. A. (1984). 'Problems, frames and perspectives on designing'. *Design Studies* 5(3): 132–6.
- Stokes, A. (1934). *The Stones of Rimini*. London: Faber and Faber.
- Weston, R. (2002). *Utzon: Inspiration, Vision, Architecture*. Hellerup: Edition Blondal.
- Whitehead, B. and Eldars, M. Z. (1964). 'An approach to the optimum layout of single storey buildings'. *The Architect's Journal* 17 June: 1373–80.
- Yeang, K. (2006). *Ecodesign: A Manual for Ecological Design*. Chichester: John Wiley.
- Zeara, A. (1995). 'Frank Gehry 1991–5, conversations with Frank O. Gehry'. *El Croquis* 74–5.
- Zeisel, J. (1984). *Inquiry by Design*. Cambridge: Cambridge University Press.