ARCHITECTURE AND COMPUTING:
a medium approach

by

RANULPH GLANVILLE

School of Architecture and Centre for Research and Development,
University of Portsmouth
Portsmouth, United Kingdom

ABSTRACT

In this paper, the argument is presented that the way in which computers are used in Architectural Design is based on and limited by our imaginings. Typically, computing is used as a tool to automate a process formerly carried out by a human rather than to expand our range of options. It is held that this is a very restricted way of using the computer and that architects and designers would be better served treating the computer as a medium, allowing it to act as a partner in design. Only then will the radical possibilities of computing become apparent, so that we may benefit from true interaction with them. Some specific limitations in our uses of computers are highlighted and ways that allow the computer to be more of an interacting partner are indicated, as are ways of exploring computing to extend the possibilities.

INTRODUCTION

This paper is based in the architectural act of design.

The essential argument presented is based in the fact that we—architectural designers—use computers as tools, rather than as media: and that by doing so we restrict their potential and their usefulness according to the limitations of our imaginings. That we use them as a factotum, to do what we might, otherwise, do (but, often, to do it faster). That while this may give rise to decent work, this work hardly begins to have a revolutionary quality of the sort claimed for computation. That, if we want to benefit from computers and associated devices and concepts (ie Information Technology), we will need to find ways of considering computers as media rather than just as tools.

For the biggest advances that new technologies have offered us, and the most significant, have been when we have found ourselves going beyond our expectations and even our wishes. When we have found that the technology has not merely done what we hoped (when we called it into being, as a tool to amplify/speed up some ability we have), but has done more, has enforced its own character and has made opportunities that we had never anticipated, never even dreamt of. When we have not limited these technologies through the restrictions of our imaginings but have allowed them to offer something back to us beyond our expectations of and plans for them (and have been prepared to—metaphorically speaking—"listen"). Then we allow that a technology has its own character: and we recognise that character, its qualities, and we acknowledge it as being more than just a tool. We accept it as what I call "a medium" (in the sense of [McLuhan 1964, 1967]).

Then, the technology, being treated as a medium, can become a participant in our (design) activities, for it is recognised as a (potential) contributor due to the separate character it is understood to have. At this point, it helps form undertakings not by amplifying our intentions but by extending—in-forming—them: in more conventional words, we could say that a medium helps turn data into information. As such, when we do something through a medium, it may return that something to us modified in ways other than we had anticipated (ie, more than amplified), as if repeated back to us by other correspondents in a conversation, as a phrase is re-presented to us in new words to help us confirm the co-respondent has "understood" [Glanville 1980, Pask 1969, 1975b], or as phrases change when whispered around a group. This can help us get beyond the limitations of our own imaginings, for we are all somehow trapped in and by the limitations of our imaginations, no matter how wonderful. The notion of a conversation, introduced by the psychologists Laing, Phillipson and Lee [Laing, Phillipson and Lee 1966], and formalised by Pask [Pask 1975b], was adopted as a paradigm for visual creativity and communication by Negroponte in the early days of the Architecture Machine Group [Negroponte 1975], with a contribution from [Pask 1975a].

Evidence for this view abounds. If we consider printing, which was intended to speed up copying (just as a simple example and also because it fired McLuhan's Gutenberg Galaxy [McLuhan 1962]), it also made redundant the illuminative art of the scribe replacing it with the new art of the typographer which, itself, spawned graphic design. And, more importantly—for this seems so particularly unrelated—it reduced the authority of the Church by making knowledge (in the form of books) widely available: knowledge was no longer associated with particular places and its embodiments were no longer the property of (under the control of) guardians (censors) [Eco 1984].

Indeed, Richard Gregory, in "Mind in Science" argues that the theories and perceptions of science are determined by the media we have at our disposal (although he calls them tools, which is confusing in terms of this discussion) [Gregory 1981]. His well-formed argument is that the whole of our science is a consequence of the instruments etc through which we explore our world and our ideas. His argument is precisely the point that tools do tend to become media.

Nearer home, and ostensibly directly part of our subject and of our time, consider Virtual Reality: we are in danger of loosing its potential power because we insist on attempting to re-create the perfectly good realities we already have (or almost have, such as walkthrough walls and mythic game spaces). We use it as a tool, to check out a synthesised simulacrum of reality, as "real" as we can make it. But, as William Bricken, the research director of the Human Interface Technology Lab in Seattle, is fond of saying: "Virtual Reality; about as unreal as you can get". His point is that a Virtual Reality that attempts to play with or at ordinary reality (or even to extend it for a mythic game) is boring and unproductive. Virtual Reality only becomes interesting and its power is only just acknowledged and beginning to be explored when we let it take us into realities that are not bent to make concessions to the consensus of the familiar reality we like to assert we inhabit (for instance, the reality of mathematics as visual space). There's clearly a message here for architects!

How we treat computers so that they limit us

Certain characteristics of our (current) use of computation are indicated below so they can be evaluated, and alternatives isolated. The qualities isolated here are presented for contrast as constituents of pairs: several have already been introduced informally.

factotum/participant

Our attitude to the computer is similar to that we have to the office perspective draughtsman. We treat it as an automatic projection maker (several presented in quick succession from successive viewing points make a walkthrough). Or as a drawing checker, to make sure changes have been consistently applied. Or as an in-house technician calculating heat loss, or quantities. Or whatever. We use it, in short, as a factotum. We define the problem, and the computer solves it within the terms we have given it. There is little or no intervention by the computer as a "partner" that might make suggestions, come up with ideas, join in a dialogue with us about the design and how to improve it.

illustration/exploration

For whatever reason (be it speed, file format, our strange and unquestioning respect for the computer…) we use computers to confirm, rather than to question or suggest. We use them to illustrate, not to explore [Glanville 1993a, 1993b], in spite of the claims that computers increase the ease with which we can generate and consider alternatives. All too often, we render a view, and that's it: it is rendered, therefore it must be good (see under "authority/question" below). The difficulty is that we do not use the computer to explore but to confirm, to illustrate decisions already made. Even such essentially "probing" programs as Virtus' "Walkthrough" have limited exploratory value, in part because of computational problems that may be overcome shortly, in part because of the television-like sucking in of the viewer.

visual/non-visual

Architecture is not just (or, perhaps, even mainly) a visual art. Yet virtually the only sense we model and output on the computer is the visual. This is worrying not only because of the narrow censoriousness (and the danger) of continuing to overplay our hands towards visual experience, but also because it fails to capitalise on a major feature of the computer: the data it stores is sense-less: we inform it towards whichever sense(s) we choose. (This is not to deny some very interesting work in this area of trans-sensory translation, such as that of Arthur Collin—based in the work of Iannis Xenakis [Muir 1989].)

super real/appropriate (suggestive)

In a similar vein, we have concentrated on the production of images that are so visually real they are super, even sur-real: so that their validity, their reality may be open to question [Lansdowne 1994]. And, further, we have come to expect this sort of presentation, which may obscure or distract from what we're interested in. The explicitness of such super real presentations removes the power of suggestiveness (part of which is that it involves us). For certain material it is about as inappropriate as can be.

authority/question

Incredible as it may seem, we are still bewitched and bemused by the computer. We continue to give credence to what the computer produces just because it was produced by a computer. We invest authority in the computer without testing or reasoning. We rarely test the output the computer gives us, or the value of the computer (and its involvement) in the first place. We sit absorbed in the computer-world, unable to evaluate, to make judgements about the value of what we are doing. It is astonishing that this continues to be the case even now.

familiar/new

Computers are, in terms of our history, new, complex and very unfamiliar. But, as the artist Paul Barker points out, interface designers have invested enormous effort in creating the semblance of familiarity and ordinariness in their interfaces [Barker 1994]. This makes the unfamiliar seem familiar, by metaphor. We talk of clicking on icons (when, being literal, we click on the mouse when the pointer is over a small pattern we accept as icon). We have desktops, file as if we had a filing cabinet, etc. Thus we blind ourselves to the new (the unfamiliar).

tool/medium

We currently use computing as a tool. That is to say, we use it not only as a tool box full of various metaphorical tools—as we specifically call them—that comprise so many programs, but as a tool in itself. We define uses we want to get from it: ways we hope it will be able to enhance our own abilities by amplification or by speeding up performance—as has been indicated above (and see, also, [de Zeeuw 1992, de Zeeuw and Koppelaar 1995]). We do not leave room for it to show the peculiar and unique qualities and characteristics it might have to offer us. We do not even treat it as having something of its own to offer. In short, we do not treat it as a medium [Pask with Curran 1982].

Many, maybe even all, tools eventually and naturally evolve into media (as defined here). A knife used not for cutting but as a pea-catapult is no longer a tool but has become a medium, as is the knife which informs the ritual of eating so that manners and etiquette appear. But this process has taken a very long time, and the power, ubiquity and universality of computers is such that we are unlikely to have that much time.

Considering the jargon of multi-media, the buzz word “interactivity”, computers are used to a surprising extent as plain tools, intended to do only what we want, not opening up new opportunities. Think, again, how little we use computers to explore, and, particularly, how little they have managed to influence the process of design and how we use computers so little in design.

Limitations in computing

We can probably viably believe that some of these limits will vanish in the fullness of time. But some (such as Baker’s observations about GUI, and Bremmermann’s limit, below) may be permanently damaging and non-reversible.

There are, indeed, “structural”, in principle limitations in computing. For instance, in using a digital approach (where there is only one basic and standard unit of difference), we trade off what in cybernetics is measured by the term variety [Ashby 1956, Robinson 1972, Glanville 1994c] for control. The digital computer gives us very precise control, but at the loss of an infinitude of variety. This is an in principle limitation (which we have ignored for so long that we have now forgotten it), as is the choice of serial processing rather than parallel (although the use of many micros and the telephone system makes, in effect, a multi-tasking massively parallel asynchronous computer, as indicated in [Glanville 1971]).

Equally, Hans Bremmermann has indicated that there is an absolute limit to computational power in the physical universe—and, of course, its constituents [Bremmermann 1977]. There are different ways of determining the actual size of this limit, but the fact of the limit remains ever present. Thus, certain computations are beyond the power of even, for instance, the earth in its lifetime, assuming it to be a solid, theoretically ideal computer. It does not take long for problems to escallate to combinatorials that exceed this limit.

There are also limits of current practicality. Speed and capacity are the two that are obvious: the difficulties of Intel's Pentium processor may well indicate a problem, also, of scale (such as indicated by d'Arcy Thompson [Thompson 1966] in showing us why mammoths could not grow to be any bigger). Probably the most serious of these concern output, which is by all accounts pretty bad even in the visual sphere. The tiny-ness of our monitor screens is specially irksome, and gives rise to a particular visual distortion: that the "surroundingness" of the visual experience of the images is destroyed and what should be surrounding us is presented only to that small cone of our vision that operates on detail: ie, the surround is treated as the focus [Bosselmann and Gilson 1993]. This is just one such failure.

And, as a final example, there are the limitations of implementation of our (computer) design choice: the use of GUI, of file formats (some of which are structured to exclude particular types of transformation), the use of luminescence rather than reflection for visual presentation, etc

How we could treat computers to liberate ourselves

We have lived with computers long enough, if not to fully understand them, at least to understand some of their characteristics. And it is these characteristics that may lead us to understand computation as a medium, and to treat is as a participant in an exploration, using appropriate (suggestive) representation and senses other than the visual, encouraging questioning and the possibility of the new: ie, to re-balance the pairings above.

Some such qualities are:

Copying

Computers make perfect copies identical to their digital originals, without deviation and in limitless numbers (our pre-occupation with computer security is a sure indicator of our sub-conscious realisation that this is so). It might even be argued that this is what computers do best.

The consequences of this prodigious copying ability may be argued to be radical [Glanville 1994c]. In brief, if everything on a computer can be flawlessly and limitlessly copied so that there is no difference between the original and the copies, the notion of ownership becomes very difficult to maintain: what is owned, and how can ownership be asserted? (This is a problem for copyright, too.) An alternative to the recognition of ownership is the recognition of origination. If there is no ownership, then anyone can do anything-they-like with whatever they find (although they should, in true academic fashion, recognise and credit origination): everything is free and everything is available for any purpose. By virtue of its copying ability, making all material potentially available to everyone to do anything with (in manners including those indicated below), the number of “interpretations” and “extensions” of original work becomes potentially very big, and the “variety” of creative interpretation becomes very large [Glanville 1975, 1988, 1994c]. This means that, by opening up our work to what are, in effect, the commentaries, elaborations and inventions of others, we increase the creativity range applied to what we probably (and erroneously) thought of as our ideas. In turn, this means that sharing is inevitable, and to be welcomed, not because of the (Inter)Net (or any network, although networks facilitate it), but because of copying. Sharing is a consequence of copying. Virtuality is a consequence of sharing [Glanville 1994a]. Hence the Virtual Studio [Maver and Petic 1994, Mitchell 1995].

Many have held this position as their dream [Nelson 1987]. Many Net users maintain this position and many believe in free software—as a matter of principle and as an assertion of the freedom of practical anarchy (how strange to have this supported by governments!). The argument summarised above explains why this in unavoidable. The computer opens up sharing through its astonishing copying ability. We may need to carefully consider the ethics of this understanding!

Copying means, in short, sharing, the mutual (and surprising, serendipitous) development of ideas.

Transformation

We are, perhaps, used to the idea of the computer as an engine for the execution of transformations. Our CAD and graphics programs are full of "tools" that are intended to allow us to transform shapes: skew, distort, cut, stretch, rotate, blend, extrude, weave, stitch, glue etc. From its earliest claims, CAD has fuelled the dream that we would soon have instant modifiability, changeability: in two words, universal transformability.

At the moment, we use such possibilities in a "weak" manner. That is, we are restrained by the traditions of our imaginations where the techniques of transformation were difficult and restrictive, and were limited by the ways our intelligences conceive and process. Now they are not so limited, it will take some time before we are relieved of these conceptual limitations, although we are beginning to open up. (Thus, we can manage those until recently unfamiliar and somewhat bizarre Bezier curves with fluency and proficiency.)

There is no doubt that our current computers are capable of transformations that were barely conceivable until very recently, even if they are still desperately slow. Yet we scarcely begin to look at the potential of the type of transformation only the machine can do.

Under this general heading we may include three types of work.

Firstly, the use of 3D Boolean operations to handle spatial manipulation and the creation of architectonic spaces: the spaces and forms that result from the various logical intersections of forms in 3D space. Some software makes this relatively easy to accomplish, for instance AutoCAD and Zoom. No doubt the operation was included in order to help the understanding of metal forms in complex junctions and such-like, but it can be used to demonstrate really interesting architectonic forms which, at very least, may extend the range of forms we can master and have at our finger tips. See, for instance, work by students at Aarhus University in Denmark, and the formal explorations of Adam Jakimowicz and Andrjez Kadusz in Bialystok, Poland [Jalimowicz 1994, Kadusz 1994]. Or software development at Cambridge University's Martin Centre under Paul Richens.

It is not that such things can transcend gravity and other sometimes tiresome reality constraints (Zaha Hadid won a UK competition for beginners to create a decent architectonic composition using ModelShop with a miraculous fragmenting space-station). It is that such forms are often beyond our conceiving: in letting the computer propose and simulate them we are also letting it re-educate us, extend our concepts of the possible and the interesting. (Nor are such forms any longer necessarily difficult to make, due to the advent of CAM.)

Secondly, morphing, blending and betweening. These techniques conceivably be done by hand (except as a sort of cartoon) because of their complexity, offer possibilities that, in part because we are not the agents carrying out the manipulations (which are thus outside our intentions) and in part because they are beyond conception, can come up with forms that are totally unexpected and unexpectable. The forms suggested are not always "practical" (although they may be). They can be adapted if they are suggestive. Students at University College, London's Bartlett School have morphed across plans to explore both the dynamic of the process of morphing (which was paralleled in their designs) and also to find plans that offered new and surprising possibilities.

As we know morphing, and blending and betweening programs nowadays, they are defined by the conditions of the originating forms as end points in the transitions. But there is no need to stop there: the processes can be extended beyond the end points. Thus, John Frazer (see below) once betweened from a Model "T" Ford to a Sierra (using his company AutoGraphics' software package "Compass"), and then on to the year 2010, to show that the proposed Ford model of that year would be almost flat: little room for people there!

Thirdly, there are generative processes such as Algorithmic Design and Genetic Algorithms. These make use of the computer's processing ability (it's hardly a computer unless it's processing). By using procedures (not necessarily precise, and possibly so complex in their interaction as to be, in principle, undeterminable), computers may be invited to propose architectures of their own.

Algorithmic Design has a following in the UK that derives from students of Paul Coates (a longtime partner of Frazer) at the University of East London [Coates et al 1995]. The aim is to create and refine algorithmic procedures that will generate building solutions to (well-defined) problems that satisfy the creator of the algorithms as being acceptable architectural responses, even as being "their" solutions (using AutoLisp in AutoCAD).

Genetic Algorithms are, of course, quite familiar by now, speeding up nature's clock and allowing artificial variety. The work of Frazer at the Architectural Association and the University of Belfast was recently on show both at the AA and on the InterNet—from which you could interact with a form-growing algorithm in real time [Frazer 1995, Graham, Frazer and Hull, 1995]. The artist, William Latham, who uses generic algorithms to generate his art, is quoted summarising an attitude I share to the use to computing in enhancing creativity thus: “With a mixture of human creativity and evolutionary systems as embodied in computer software, one can produce extraordinary things, things that are beyond the human imagination. There comes a point when you cross the boundaries of what is familiar. The machine comes back with surprising results and you suddenly find yourself thinking, gosh, I’ve never seen things like this before” [“Breeding Artist”, Life Magazine, The Observer, London, Sunday March 12 1995.] Note that his computer is his partner in this enterprise.

In summary, transformation can be of (at least) three types:

- transformational spatial manipulation (going beyond the limitations of familiar human skills such as skewing etc, to exploit transformations that exceed our current visualising skills—such as 3D Boolean operations)

- blending, morphing and betweening (finding spaces and forms that are the result of a partial, productive mix of two origins, possibly extrapolating beyond either)

- algorithmic/procedural design (the generation of form and its consequent space through the application of (developing) rules)

Abuse

In these days of correctness, there is a danger in using such a term. Nevertheless, it reflects a naughtiness, risk and incorrectness that I wish to capture and have argued strongly in favour of [Glanville 1992]. For the overriding notions that attach to the term abuse, in this context, are of doing the risqué thing: naughtily using software in a way it was never intended to be used.

Most (all?) software has a fairly clear purpose: a set of tasks it is intended to facilitate the solution of. But most software can also be used in other ways than those intended. And, often, using the software in this manner turns out to be fruitful, worthwhile and fun.

This alternative use is what I call software abuse: to use it beyond and outside the scope intended for it. But it is possible and may be surprising and rewarding.

Where software can be so abused, we can learn something about the software and the computer-as-medium (and our “perverse” imaginings!). For, in finding things to do that are outside the task, we are discovering, also, the unknown and the unexpected in the computing environment: that is, we are discovering what the computer can offer us "back", beyond our demands. This is, in the terms given above, the difference between a tool and a medium. While all the qualities indicated help us understand, through careful consideration and then examination, what the qualities of computing are—and hence how we can consider it as a medium—abuse, by going direct to the unexpected, may open the gates of "mediumship" directly to us.

Examples of techniques that involve software abuse include the following:

- compression: for reasons of economy and of speed we use compression. Lossy compression of images (eg JPEG), applied repeatedly, can lead to the emergence of surprising forms through the obliteration and the centralised grouping of (visual) data. This is abuse, for the intention in lossy algorithms is to minimise the distortions of the compression.

- colour palettes: a technique similar in spirit to compression, perhaps. By the reduction (eg) of a picture from 24- to 8-bit colour, and the substitution of alternate palettes or of some colours within the palette, we can radically alter an image. Rolling colour changes animate such images in extraordinary ways. This qualifies as abuse because the purpose of 8-bit substitution for 24 is to reduce file-size with minimum change to the image

- auto-tracing: auto-tracing chooses changes in colour and contrast to generate contours or outlines. It is essentially unpredictable (except when using highly specialised programs such as Adobe's "Streamline"). Auto-tracing a drawing produces unexpected shapes, often of the sort we wish we could make but don't really know how to draw. Framegrabbing from video images of a site, for instance, and auto-tracing from these images can give rise to forms from and with which to design, enriching our formal vocabulary. The technique can be applied to images obtained in any way, for instance by compression or colour-depth down-grading.

Students of mine (in the UK and in Poland) have used this technique to generate designs and, although the designs are not always very distinguished, that can just as well be attributed to lack of familiarity and practise, and to the continuation of the tradition in which the less designerly students find a refuge in computing, as to anything else.

- Scanning: Co-op Himmelblau used computers in their recently completed Museum in Groningen, Holland (see report in Building Design, January 27, 1995). Constructing some highly abstract compositional drawings, they scanned these into the computer, overlaid them, and connected vertices. The resulting forms were transmitted to a CADCAM station in a shipyard, where the forms were constructed and shipped for final assembly to the site (on a canal). A similar technique has also been excitingly used elsewhere by students constructing small "conceptual" models which are scanned into the computer, thus releasing their forms to the computer. Here, the computer's ability to connect, to understand intersections, to find edges is used to generate unpredicted forms that transcend and extend the imaginative limits of human mind.

It will be noted that the techniques given above lead to the generation of new and surprising forms from graphic material (over which we do have control) through our willingness to hand over some of that control to the computer: we let the computer take control, make decisions and carry out actions, on its own terms, that we would find hard to do.

This short list is awaiting expansion: it is limited by the imagination of the Author and those he has asked for suggestions, amongst other things. It is just a beginning, and perhaps a somewhat quirky one at that. But the point is that it is a beginning, and it begins to open up new possibilities which result from qualities of the computer and its software.

Conclusions

It seems clear that computation can be used in ways such that it can be treated as participating and offering us possibilities that we could not, of ourselves, imagine: and developing with us a sort of productive conversation when treated as having its own qualities (on top of those programmed in) and as "kicking back at us"—ie, computing can properly and actively be considered as a medium.

In this respect computing can reflect, more accurately than otherwise, the designer's back-of-envelope design stages, which are frequently held to be where and how ideas originate. Until computers can participate in this "stage", they will not be fully participating in creativity, and will not be truly helping us expand our creative potential and powers.

To achieve this involves a number of modifications and extensions to our approach, many of which have been highlighted above.

These include:

- having (taking) less control ourselves (allowing the computer to take command) so that we can be serendipitously surprised;
- opening up our minds so that we can become aware of what is being offered us by the computer;
- joining in a productive "conversation" with the computer as a participant. Conversations allow us to develop ideas by allowing others (eg, computers) to join in our working;
- as a result of copying, sharing—recognising origination rather than ownership, and allowing the borrowing and development of work by others (and of others by us);
- developing algorithms and transformations that transcend the types of performance we can even envisage, ie allowing performances that only the computer can manage;
- allowing the suggestion of shapes that we may like but which are beyond those we can normally draw (CAM has released us from the production tyranny of the rectangular: it is now time to study and work in "shape" and not its idealised abstraction, geometry);
- recognising that computers deal with sense-less data: we can take this abstract structure and translate it into and across whichever senses we wish;
- learning to explore and ask questions with computers, to look for the new and not the already recognisable, rather than those set by the limitations of our imaginings.

Of course, many of these qualities are just the qualities we expect in the creative regardless of how or in what medium they exercise it, and which we encourage students to acquire if we wish their latent creativity to show through and develop in a potent manner.

Naturally, there are problems. Navigation in the HyperSpace of work and idea sharing is difficult, for one thing (but see [Glanville and Ferris, forthcoming] for a solution). If abusing computation is another approach, it needs a wider, more systematic handling, application and analysis. It is a large research programme, transcending many fields (in some of which we have little quality experience). We are only at the beginning.

But what is clear is that, in treating computation as a medium, we have a better chance of gaining more benefit than if we continue to treat it as a tool.

And what about media other than computation?

References

Ashby, R, Introduction to Cybernetics (London: Chapman and Hall, 1956)

Barker, P, “Hype, Hope and Cyberspace –or– Paradigms Lost” The Virtual Studio, Proceedings of eCAADe 1994, edited by Maver, T, and Petric, J, (1994)

Bosselmann, P and Gilson, K, “Vizualizing Urban Form” in Aura, S, Alavalkama, I and Palmqvist, H, editors Endoscopy as a Tool in Architecture (Proceedings of 1 EAEA) (Tampere: TUT School of Architecture, 1993)

Bremmermann, H, “Complexity and Transcomputability” in Duncan, R and Weston-Smith, M, The Encyclopaedia of Ignorance (Oxford: Pergamon, 1977)

Coates, P, et al Generative Modelling Workbook (London: Centre for Architectural Computing, UEL, 1995)

Eco, U, The Name of the Rose (London: Chivers, 1984)

Frazer, J, An Evolutionary Architecture (London: Architectural Association, 1995)

Glanville, R and Ferris, C Pictures at an Exhibition (Systems Research, forthcoming)

Glanville, R, “CAD Abusing Computing” Proceedings eCAADe 1992, edited by anon, (1992)

Glanville, R, “Exploring and Illustrating” Proceedings eCAADe 1993, edited by Smeltzer, G (1993a)

Glanville, R, “Looking into Endoscopy” in Aura, S, Alavalkama, I and Palmqvist, H, editors Endoscopy as a Tool in Architecture (Proceedings of 1 EAEA) (Tampere: TUT School of Architecture, 1993b)

Glanville, R, “Remoteness and the Value of Sharing” The Virtual Studio, Proceedings of eCAADe 1994, edited by Maver, T, and Petric, J, (1994a)

Glanville, R, “Representations Fair, Honest and Truthful” Materialy Konferency Jne, Projektowanie Wspomagane Komputerowo—Bariery i Inspiracje Twórcze edited by Kadysz A, (1994b)

Glanville, R, A Cybernetic Development of Theories of Epistemology and Observations, with reference to Space and Time, as seen in Architecture (Ph D Thesis, unpublished) (Uxbridge: Brunel University, 1975) also known as The Object of Objects, the Point of Points,—or Something about Things

Glanville, R, Architecture and Space for Thought (Ph D Thesis, unpublished) (Uxbridge: Brunel University, 1988)

Glanville, R, The Architecture of the Computable (Design Studies Vol 1 No4, 1980)

Glanville, R, Using Electronics in the Environment—Free and Uncensored Video through Localised Cellular Networks of (Pinhole) Video Cameras: Design Thesis at the Architectural Association School, London, 1971

Glanville, R, Variety in Design (Systems Research Vol 11, No 3, 1994c)

Graham, P, Frazer, J, and Hull, M, "An Application of Genetic Algorithms to Design Problems with Ill-Defined or Conflicting Criteria" Problems of Values and Invariants (Proceedings of the Conference on Problems of Values and Invariants, 1993), edited by Glanville, R and de Zeeuw, G, 1995

Gregory, R, Mind in Science (Harmondsworth: Penguin 1981)

Jakimowicz, A, "Abstract Modelling—Forming and Exploring" The Virtual Studio, Proceedings of eCAADe 1994, edited by Maver, T, and Petric, J, (1994)

Kadusz, A, "CAD the Tool" The Virtual Studio, Proceedings of eCAADe 1994, edited by Maver, T, and Petric, J, (1994)

Laing, R, Phillipson, H and Lee, R, Interpersonal Perception (London: Tavistock, 1966)

Lansdowne, J, "Design and the Virtual Studio: Some Remarks on the Impact of Computing in Design" The Virtual Studio, Proceedings of eCAADe 1994, edited by Maver, T, and Petric, J, 1994 (loose bound insertion)

Maver, T, and Petric, J, The Virtual Studio Proceedings of eCAADe 1994 (Glasgow: University of Strathclyde, 1994)

McLuhan, M, and Fiore, Q, The Medium is the Massage (Harmondsworth: Allen Lane 1967)

McLuhan, M, Gutenberg Galaxy (London: Routledge and Kegan Paul, 1962)

McLuhan, M, Understanding Media (London: Routledge and Kegan Paul, 1964)

Mitchell, W, City of Bits (Cambridge, Mass: MIT, 1995)

Muir, M, Performing Space—Space and Music (unpublished Dissertation) (Portsmouth: School of Architecture, UOP, 1989)

Negroponte, N, Soft Architecture Machines (Cambridge Mass: MIT, 1975)

Nelson, T, Project Xanadu (version privately circulated on computer disc, 1987)

Pask, G, Aspects of Machine Intelligence, in Negroponte, N, Soft Architecture Machines (Cambridge Mass: MIT, 1975a)

Pask, G, Conversation Theory (London: Hutchinson, 1975b)

Pask, G, The Architectural Relevance of Cybernetics (Architectural Design, September 1969)

Pask, G, with Curran, S, Microman (London: Century, 1982)

Robinson, M, Classroom Control—some Cybernetic Comments on the Possible and the Impossible (Instructional Science Vol 8 1972)

Thompson, D'Arcy W, On Growth and Form (abbreviated by Bonner J) (Cambridge: CUP, 1966)

Yakeley, M, and Coates, P, "The Virtual Chings Head" The Virtual Studio, Proceedings of eCAADe 1994, edited by Maver, T, and Petric, J, (1994)

Zeeuw, G de, and Koppelaar, H, "Object, Systems and Invariants" Problems of Values and Invariants (Proceedings of the Conference on Problems of Values and Invariants, 1993), edited by Glanville, R and de Zeeuw, G, 1995

Zeeuw, G de, Soft Knowledge Accumulation or the Rise of Competence (Systems Practice Vol 5 No 2, 1992)