

---

**Chapter 16**

# **How can CAD provide for the changing role of the architect?**

**Simon Ruffle**

At the RIBA Conference of 1981 entitled 'New Opportunities', and more recently at the 1984 ACA Annual Conference on 'Architects in Competition' there has been talk of marketing, new areas of practice, recapturing areas of practice lost to other professions, more accountability to client and public 'the decline of the mystique of the professional'. It is these issues, rather than technical advances in software and hardware, that will be the prime movers in getting computers into widespread practice in the future.

In this chapter we will examine how changing attitudes in the profession might affect three practical issues in computing with which the author has been preoccupied in the past year. We will conclude by considering how, in future, early design stage computing may need to be linked to architectural theory, and, as this is a conference where we are encouraged to be outspoken, we will raise the issue of a computer-based theory of architecture.

The first issue is concerned with the design process itself. How much process is there in design? On the one hand, architects want freedom to solve design problems in an unstructured, unconstrained way; on the other, if computers are to be involved, some definition of the processes and data flows in design would be of help. There seems to be resistance among architects to set consistent standards and explain their methods. Yet computer people are not the only ones that would like architects to be more explicit about their methods. Many within the profession feel that it is only by explaining the process of design, its inputs and its outputs and its particular qualities that potential clients and the public can be brought round to seeing the architect in a better light, to understand what design is really about. This trend towards architects defining their methods for themselves will help them not only to put themselves over to the public better but also to appreciate in what ways computers could be of benefit.

The second issue is that computers cost a lot of money, and this is unlikely to change. Architects are traditionally unused to large capital investment in equipment, predicting future workload and committing themselves to growth like a commercial company. They prefer to think about quality of product rather than efficiency of process. Unless these attitudes change fundamentally the cost of computerization is always going to be a problem. However, attitudes are changing. Architects are being encouraged to market their services and compete for clients. Advertising is allowed. One practice has successfully issued shares on the Over-the-Counter Securities Market. Computers are a marketing tool, a part of an improved service and can be afforded and justified if the right commercial approach is taken.

The third issue concerns new techniques. Computers offer architects new techniques such as environmental simulation, project management

packages, colour visualization and early cost analysis. A familiar reaction to computer programs such as these is 'very clever, but our practice does not do much of that'. Architects are being encouraged to widen the scope of their services into new areas, and to reclaim lost work from allied professions. If surveyors and contractors are buying drafting systems, why shouldn't architects buy costing or project management systems?

As these trends cause architects to make more use of computers and, as is happening now, vendors begin to develop more sophisticated products in response, attention will turn again to the old battleground, the early design process, where more than once CAAD has been told it has no place.

The author is particularly interested in how much the acceptance of a design-aid computer program depends on the relationship between the methods of the program and the theory of architecture adopted by the user. The OXSYS program (Hoskins, 1976), for example, was firmly rooted in the International Style (defined by Reyner Banham (1960) as 'mature modern architecture of the mainstream'). The language of the style was well defined: grid planning, cores, skeleton frames, component system and flat roofs. There was a passion for orthogonal composition. Le Corbusier wrote,

... thanks to the invention of reinforced concrete which offers us the most pure mechanism for orthogonal composition.

With such ideas as Sullivan's 'form follows function' and Le Corbusier's 'the plan is the generator' still current in the early 1970s it is not surprising that we should find Paul Richens writing,

The starting point of OXSYS/BDS is a brief... some breakdown of the building into functional zones ....'

or Ed Hoskins,

The design of a building system is to a greater extent involved with designing the logic controlling the assembly of sets of components, dimensional coordination, and these are the rules that govern the design of computer applications.

UK architects pondering whether to buy OXSYS (now BDS) in the late 1970s were perplexed (Ruffle, 1979) by the program's inability to model other than orthogonally planned buildings. In retrospect this appears less of a technical issue and far more a consequence of the architectural theory prevalent at the time of the software's inception.

This presents a problem to the software designers of today. Architects do not seem to have in common any well-defined theoretical basis for design. Architectural education promotes the fashionable 'design guru', the person who appears to be good simply by some magic chemistry which most architects cannot have and which certainly a computer can never have. Often the guru-person is a talented architect, but the reasons why their particular way of designing is good are not brought forth. Rather than being taught, student architects are expected to divine how to design, so no wonder practitioners find it difficult to evaluate how useful a computer might be in aiding their design process.

CAAD must focus future development on the creative design process. It

is not sacred territory and, as seems necessary, we may have to underpin work in this area with our own corresponding contribution to architectural theory. Research into computer-oriented theories of architecture has been understandably covert. In the near future simple commercial pressures from both vendors and users wanting to widen the scope of the computer in architects offices may welcome such ideas back into the open.

### References

- BANHAM, R.** (1960). *Theory and Design in the First Machine Age*, London: Architectural Press  
**HOSKINS, E.** (1976). *Integrated Computer Aided Building and the OXSYS Project*, Applied Research of Cambridge Ltd  
**RUFFLE, S.** (1974). *Computers in YRM*, London: YRM Architects

(The opinions expressed in this paper are the personal opinions of its author. They do not necessarily express the views of any company.)