AN EDUCATIONAL STRATEGY FOR CAAD AND ITS IMPLEMENTATION IN A NEW SYSTEM WITH A SOPHISTICATED

DERMOT O'CONNELL
Professor of Architecture
University College Dublin
School of Architecture
Richview, Clonskeagh
Dublin 14, Ireland

ABSTRACT: Reasons are examined for the slower uptake of CAD in architecture than in engineering. An appropriate response to Computer-Aided Architectural Design is overdue from the educational sector. Schools of Architecture should put CAAD to the forefront in their plans, taking an ambitious long-term view and aiming for high-quality system design to anticipate the industry's drive to produce first-class equipment progressively more cheaply. Schools should press for changes in the way facilities are supplied to them. They should discard obsolete software, buy software commercially, stick to what they can do best, and plan for concomitant changes across the curriculum. A new CAAD system with the emphasis on the design interface being implemented in UCD School of Architecture is briefly described.

INTRODUCTION

The profound implications of automation for architectural design have been recognised for more than twenty years and the literature abounds in attempts to put computing at the architect's service. It must be acknowledged however, on the one hand that the rate of 'uptake' in architecture of available information-processing techniques has been slow, and on the other that the development of true design aids has not been as easily realised as promised a generation ago. The latter applies even to engineering, though with less force, despite the conjunction of production and design in engineering practice (as for example in the aerospace industry) which conduces to investment on a scale inconceivable in architecture as it is practised now.

In the past few years the rate of uptake of CAD in engineering has been rapid and CAD is now becoming a major part of courses in engineering schools. It is timely therefore to consider how an architectural school should respond to the near certainty that within ten years CAD systems will be commonplace in architectural practice. Our point of view is that an educational institution must anticipate changes in the way a profession works but not so much drill its students in the accidental and arbitrary manifestations of new technology, as educate them in principles that will see them through their working life and give them the capacity to adapt constructively to change. What we see of the computer revolution from this vantage point is not restless innovation but relentless obsolescence, with which we have to come to terms.

Definitions

We start with the assumption that, ultimately, through embedding CAD in our method of working we will change our conception of man-made objects and the art of their production. Quite how we will change our concepts it is unnecessary to say here but it is necessary to justify putting CAD in an educational setting on a completely different plane than other design aids such as photography and model-making.

The slow development of CAAD systems, indeed, is partly explained by the difficulty of deciding what is meant by architectural design. Some professions implicitly define design(ing) as a process that only designers can carry out. Others use the word design quite indiscriminately: sometimes to denote the activity (of making drawings, for example); sometimes for the drawings themselves; and sometimes for the building actually built. In these senses one could design a design for a design. Because the criteria by which an architectural work is judged are hard to specify and the frame of reference may differ for different observers at different stages of production, when we use a CAD system to evaluate performance (in the broad sense) of something we are designing, we must be clear what kind of model we are testing explicitly or implicitly.

We prefer to adopt a working definition that is concrete rather than abstract

> A design is a set of instructions for the production of a new (object or) building
> Contriving this set of instructions (by whatever mans) is designing

The set of instructions should be sufficient, explicit, coherent, consistent and feasible, according to professional and industrial conventions of production. Computer aided architectural design(ing) is therefore the making of sets of instructions, whose subject matter is architectural, with the help of a computer. We use the computer to externalise (make explicit) design geometry, for example, and to test its consistency and feasibility.

According to this definition a (computerised) drafting system is a CAAD system although critics could fairly say of many CAAD systems, so called, that they are neither truly architectural nor truly design systems. We prefer, however, to retain the generality of our definition because it is unprofitable otherwise to have to say once and for all, where drawing stops and designing begins. There are two other reasons for preserving this generality. Firstly, even according to the conventional understanding of design(ing), an automated drafting system, though limited in conception might be used quite creatively - which does not preclude our saying that some CAAD systems are more design-orientated than others. The 'quality' of the interaction, actual and potential, between the user and the system is of crucial importance, and merits fuller discussion later. Secondly, the generality of our definition allows us to see the activity of designing, whether mediated by computer or not, alongside the architect's spectrum of information-processing tasks. Designing and information processing are not the same thing. Information processing is used to assist in making decisions (instructions). It does not make decisions (in our frame of reference) although it may appear to give designers little latitude (i.e. to make decisions for them) when it teases out the constraints (or inconsistencies) latent in a commission.

BARRIERS TO THE ADOPTION OF CAD

There are various reasons, which one could describe as technical, cultural and philosophical, for the tardy adoption of CAD in any comprehensive sense in schools of architecture. The technical reasons can be considered almost independently of the others: the unavailability until recently of suitable systems (of hardware and software) at the right price. Computer manufacturers are responsible for producing suitable hardware, and although the lack of suitable applications software can be ascribed to shortcomings in architectural research (of which more later), the absence of good hardware is sufficient explanation for the shortage of good CAD systems in architectural education.

To be appropriate to architectural use the hardware must facilitate those methods of working that are peculiar to architecture and those evolving methods we can reasonably foresee. Until the technology of display screens neared its present level it was arguably premature to promote CAAD outside the research community. Piecemeal applications on unsuitable equipment may have discredited the movement in architectural computing, although such setbacks could hardly be peculiar to architectural education. It is a technology characterised by the need for urgency and patience at the same time.

Our broad definition sees the work of the architect as making decisions about subject matter of a particular kind. Some CAAD systems are not truly architectural for a very good reason - because they were designed as drafting systems for engineering. It might of course be contended that a drafting system whose end product, say, is a line drawing, should be indifferent to whether an architect or engineer is using it. Every drafting system will draw straight lines, make curves, dots, circles, etc, and it is easy to conceive a system equally at ease with engineering and architectural graphics. But the criteria of efficiency and fitness suggests that if the geometry of engineering differs from that of architecture, the same graphics system will not serve both well. The geometry of mechanical engineering is the geometry of cylinders, pipes, gears, shafts, springs, threads and extrusions. The geometry of architecture is the geometry of building: laminae, surfaces, prisms, windows, doors, corridors, cills etc. The operations one wants to perform on the graphic elements will for the most part differ in frequency apart altogether from the operations on the geometrical model and its data base.

We should summarise the possible technical barriers as follows:

System: Poor system design and integration
High costs to buy and maintain

Hardware: Unsuitable for subject-matter and users; unnatural or clumsy interaction
Poor performance: deficiencies in speed, capacity, resolution, extensibility, efficiency, consistency of response, reliability etc.
Special demands - (air conditioning, darkened room etc); poor ergonomics, poor communication facilities

Software: Piecemeal and uncoordinated
Unclear educational/didactic purpose
Poorly thought out theory
Limited capacity for innovative use and development
Not portable
Obsolescent operating system or language
Difficulty of linking to other software modules or database
Poor documentation
Absence of standards

Cultural Barriers

Much resistance comes from the architect's fear that he will be less important (in the production of building designs) if he yields ground to automation. He may maintain that design by computer (which is what he understands by CAD) is impossible and bad. Or he may say that it is bad even if it is possible. What he probably fears is either that the profession will lose work and influence to 'laymen' (or even clients) using CAD, or that within the profession the adoption of CAD will lead to 'a lower standard of design'.

These are not unreasonable fears. Every profession has its strains in adjusting to automation, as it has had in adjusting to mass literacy. Information systems do not need to recognise professional boundaries. Automated system for example, are available in medical practice, which when clinical information is supplied, enable ailments to be diagnosed with greater certainty than by ordinary medical practitioners. The doctor's response might be to argue that the system is not practical in widespread use, or that the strength of the profession lies not in diagnosis but in therapeutics and research. All we would say here is that it is hard to set limits to the knowledge-base on which any profession is founded and that the accelerated processing of information cannot impair any professional function that is not information processing in some form already.

As some information-processing functions are elaborated within the building designer's role, it does not follow that the residual functions are contracting in extent or importance. The contrary indeed may apply - the designer gains leverage over a larger knowledge base.

In an educational setting the resistance to CAD (apart from the short-comings in hardware and software) comes therefore from professional fear, compounded by a failure to think through the role of information in tile process of designing. This failure will of course be more damaging where the Beaux Arts spirit holds sway.

In an educational setting there will also be two conservative forces - one from architectural teachers and one from the institution itself. Firstly, practitioners immersed in commerce may dominate the intellectual and professional formation of students whose educational needs are a generation removed from their own. Secondly, university institutions themselves, are not necessarily receptive to innovations such as CAD even where schools of architecture might be. Universities have tended to follow a definite pattern in the evolution of computing services and in the allocation of resources.

Apart from architecture in which it has been afforded a certain mystique, designing has never been given a place of much importance in university education. Engineering design has always featured in the curriculum but close scrutiny suggests the term has generally been used for courses in analysis with little claim to synthesis. In default of the emergent methods of analysis and computation, designing in any comprehensive sense was something best learned in practice, over time. Engineers have had an advantage in their conversion to CAD, because their problems tend to be well defined even where hard to solve, because they have developed the tools, because of their robust tradition of numeracy and because of their pragmatic concept of design.

In the past generation CAD has become not just the means to computerise engineering, but also possibly the means of bringing the concept of designing right to the forefront of education. Nor is there any aspect of architecture that artificial intelligence (of which CAD is a special case) will not touch.

Unfortunately however, universities are slow to see the change in user requirements and if they do they may find it most inconvenient. Consider the transition in the structure of academic computing facilities from:

(a) Central processor for numerical computation using batch processing, card input, printed output, to

(b) Time sharing, satellite terminals, interactive languages. Vdu displays, text processing, plotters. Even the latter as a way of organising computer services is essentially bureaucratic - meant not pejoratively but in the sense that it is large, centralised, and difficult to change.

The operating systems besides, and software provided in mathematics, engineering and statistics, will reflect how computing needs have been perceived for thirty years - analysis. The peripherals, the operating systems, the interfaces will be unsuitable for design aids, and the design schools will be under-represented in configuring the services.

Just as important is the professional fear of educators that in the new order traditional teaching methods will be superseded by methods they cannot control . As 'expert systems' come to be integrated into CAD systems the structure of teaching will be threatened as much as professional structures are now. Consider an expert system for the structural design of buildings. The professional question is, if one can master the expert system itself, how much must one know in detail about structural theory? The academic question is whether and how far an engineer is still necessary to teach structural theory to students of architecture.

It is also necessary to ponder the possible effects of specialised software which allows an expert to move his expertise into the memory of a computer. Donald Michie, Professor of Machine Intelligence at Edinburgh University has recently announced such a program, working by 'inductive learning' rather than by dialogue as in traditional programs.

Philosophical Barriers

A consensus does not exist delimiting the criteria by which an architectural design can be evaluated. As soon as the designer provides the answers the critic is liable to change the question.

Many aspects of a building's performance can be predicted piecemeal, and related to user satisfaction, but there are problems of synthesis and of balance. And even where the physical theory is adequate and the psychological basis linking this to user satisfaction is understood, there remain philosophical difficulties about the whole concept of user satisfaction.

In some cases the physical theory is inadequate. In others the physical theory is satisfactory but its significance is only established by arbitrary regulation. Daylight distribution is trivial to predict from the geometrical description of a building. But what do we make of a particular, result apart from compliance with an arbitrary code?

In other cases the physical theory is adequate but too complex to follow to a conclusion of use to a designer. Only recently for example has it become possible to simulate the dynamic thermal behaviour of buildings in any detail.

These problems belong in the domain of architectural research. They may inhibit the development of CAAD in a sense, but we firmly believe that the adoption of CAAD will give an urgent reason for tackling them, and in some cases the means also.

AN EDUCATIONAL STRATEGY

We now try to articulate the kind of strategy we think appropriate for a school of architecture. The first component is a policy on the provision of CAAD facilities, the second on the harmonisation of academic courses with CAAD and the third on research and development.

CAAD Facilities: Hardware

Whatever facilities are provided should we believe, recognise the following industrial trends:

(a) the continuous introduction of faster and more powerful processors, at progressively lower prices

(b) the progressive improvement in the resolution of raster-graphics screens with local intelligence

(c) the availability of colour screens of good resolution at reasonable cost

(d) the availability of very large data bases commercially within each profession

(e) relative stability in the cost of peripherals such as digitisers and plotters

Our experience suggests that a complete drafting system can now be produced for between a quarter and a half of the cost (according to the choice of peripherals) of a system of comparable performance two or three years ago. We think it is now widely accepted that for graphic computing a dedicated processor is necessary, but we do not rule out its use for other, even simultaneous jobs, provided graphic processing which is the most demanding computationally, is given pride of place. The software, besides, should be contrived to compute as much as possible (such as display functions) locally, to reduce the load on the host.

There is never an ideal time to buy hardware. By the time a system is delivered it is likely to be obsolescent and over-priced. This does not make one a fool for buy it. All one can do is buy a sufficiently good system in particular circumstances, paying due regard to trends in hardware. If it is to be replaced in a few years, the important thing is whether it will have the same operating system.

The resolution of VDU screens should be of the order of 1024 x 768 as an absolute minimum. Raster-graphics screens of superior resolution are becoming available cheaply and in a few years will be standard for graphics work. It is best therefore to be as ambitious as one can afford, and not to shackle one's system to screens of poor resolution.

In some types of work, (for example where the screen is mediating elementary operations an stored drawings), it is the resolution of the plotter (which is normally high) that is crucial. But in an architecture school the greater weight of designing rather than production (as in an office) justifies emphasis on a high graphic quality in the screen image. To persuade the sceptics also it is necessary to make the design interface as attractive as one can afford, and, to provide selective erasure on the screen is absolutely mandatory.

We are not convinced of the necessity of colour graphics for ordinary design education, but we think a CAAD system should be structured to allow such facilities to be added, when the technology and software are available at the right price.

Our proposals for hardware could be summarised as follows: to acquire as far as one can afford;

(a) a large 16-bit (or 32-bit) processor (multi-user and multi-tasking) in a product line that facilitates painless 'stretching' in the future, as prices come down

(b) large-resolution graphic screens with provision for future upgrading, with improvements in raster and colour technology

(c) plotters, digitisers. keyboards and terminals according to demands on workstations

Note that there is no place in our scheme for 8-bit microcomputers, nor for low-resolution graphics nor for simple storage tubes.

CAAD Facilities - Software

Because investment in software is likely to exceed that in hardware for the foreseeable future, it is necessary to plan for the development or acquisition of software over the long term, and to take software costs seriously.

At the risk of heresy we assert that it is not the business of educational institutions (with very limited exceptions) to write applications software. We hope that in architecture, the situation will not long persist where the needs of the architectural user are so poorly understood and the market so poorly structured that the architecture school cannot purchase CAD packages off the shelf as engineering schools can.

We think that every school of architecture already in CAD should have its existing software assessed for suitability in use over the next five years. To compound our heresy, we assert that much software written over the past ten years, without the benefit of today's equipment and software tools should be discarded.

But our approach is based on the assumption that it is premature to attempt CAAD in a comprehensive sense. We know too little of the psychology of the interaction of designers with CAD systems, and too many theoretical questions remain unsolved linking the definition of a building to its performance.

Our grand design is a broad outline for a CAAD system of which the first working part is a drawing system, whose data structure and functionality are meant to facilitate other CAAD functions as experience is built up with the drawing facilities. The drawing system is the organising principle for every thing else and we would like to feel we have contrived it not as a draftsman's drawing system, but with the feel of a sports car rather than a grocery van. We have put much emphasis on speed of interaction and flexibility and conceive two levels of skill in use - the beginner (or draftsman) and the expert level, where experienced users can take short cuts, and, we hope, use the system creatively with a sense of discovery. It is for others to decide whether we have succeeded.

CAD for Professional Use

The emphasis in professional practice on production and in education on learning does not justify a difference in CAD software itself but may justify greater spending in the former on output devices, such as plotters, and in the latter on devices such as screens.

A professional system will also require much back-up storage for drawings and ultimately a very large store when professional data-bases become available. A school is not meant to duplicate professional practice and does not need facilities on the scale of practice. At the saw time its CAD facilities should be compatible with those of practice. The school should use its CAD facilities not just as a design tool for its students, but as a means of learning about CAD itself and acting as an 'innovative customer'.

Academic Courses

There are two ways of considering CAD in a school of architecture. One is to simply provide a CAD facility as a tool and to allow students to produce designs with its help, without any study of computing theory or any experience of writing programs.

To use CAD as a tool is we think a perfectly defensible educational policy. It does not preclude valuable insights into methods of designing or into the theory behind different academic subjects. In the short term the extensive use of CAD systems limited to drawing functions may be the best way to change the climate of opinion in favour of CAD generally. And it may be more productive to achieve limited objectives than to have greater ambitions with ambiguous results.

In the other approach we would consider a broad study of computer science preferable to the piecemeal writing of applications programs. Experience of programming is important, but it can be accommodated in an introductory course in Computer Science. The cult of computer programming in the wrong context has led to a great wasting of resources.

It would be shortsighted to ignore the possibilities offered by CAD to improve the teaching of almost every subject taught in an architectural course. Structural engineering, mathematics, environmental science and building technology for example would benefit from the facilities for simulation and pictorial representation. Other subjects would benefit from access to suitable data bases. This is apart from techniques of Computer Assisted Learning which would be enhanced by forms of interaction beyond the scope of conventional CAL.

CAD in the University

The provision of CAD in a school of architecture (or engineering) requires a computing system that is contained within an individual school, contrary to practice in most university computer services. This is dictated not by departmental pride but by the nature of CAD. The nature of CAD also requires applications software of a kind that it is not, in our view, the function of universities to write themselves. The university must therefore accept the responsibility of;

(a) supplying the necessary hardware for the school's exclusive use
(b) purchasing the necessary CAD applications software
(c) arranging maintenance and support for hardware and software
(d) providing software expertise below the applications level
(e) providing academic staff
(f) providing advisory services
(g) providing linkage to central services

AN IMPLEMENTATION OF CAAD

Our aim has been to sketch out a coherent plan for CAAD and to implement it in an educational setting with hardware that would make it equally suitable for use in the architectural office. We plan software development (or acquisition) around the modules that initially constitute our drawing package as follows:

DRAFTING:	Working Drawings - production and editing
	Sketch Design Aids
	Visualisation Techniques

ANALYSIS AND SIMULATION:	(of heat, light, sound phenomena, topology etc)
OFFICE MANAGEMENT:	Text editing, filing etc
DATA BASE MANAGEMENT:	(for architecture and building data base)
COMMUNICATIONS:	Electronic Mail, sharing of resources etc

Hardware - Processor

We considered the production and editing of working drawings to be the most significant task both in terms of demands on the processor and also in terms of usefulness to the end user. This has largely determined the hardware specifications of our system.

In its implementation we felt that there would be a number of advantages if certain modules could be run in parallel - if, for instance, the interpretation routines, which require large amounts of CPU time with little or no I/0, could run apparently simultaneously with input procedures requiring relatively little CPU. The same applies to features such as background plotting.

In addition, being able to run office management routines on the same processor seemed attractive particularly if the terminal could be placed in an existing studio (and thus minimise disturbances to existing work practices) rather than have to be placed physically near the processor as in a stand-alone workstation.

The other major criteria were cost and the availability of extra memory. In order to be able to support the widest range of graphic screens and still maximise the flexibility of the system we decided to use an in-core data base. Hence the processor should be able to support at least .25 mbytes of memory per task in addition to that required by the operating system. On the subject of cost it was felt that a multi-user system offered more flexibility and better value for money than the single-user equivalent, especially for the smaller office.

Digital's PDP-11 range of computers seemed ideally suited to our needs. As well as offering a wide range of peripherals a large body of software is currently available for these processors. The ultimate choice of model and actual configuration (whether network of smaller units serving 2 or 3 workstations or a larger say 11/70 serving 10 or more) has yet to be decided but our development system is based on an 11/23+. This, we think, will serve at least 2 graphic workstations, and is upward-compatible with the VAX range.

Hardware - Screen

The choice of screens is largely personal. It was our aim to be able to support the widest possible range, the most crucial requirement being some degree of selective erasure, not primarily for the deletion of individual lines but more for the display of alphanumeric prompts etc.

Our own choice was a Textronix 4114. With a type of graphics that is largely line-orientated, we felt able to exploit its local intelligence to full advantage.

We also expect a large number of cheaper 4110 series-compatible raster screens to be on the market within a short period of time.

Hardware - Miscellaneous

It is generally proposed to use thumb wheels for input with the option of a small 11"x11" tablet if desired.

Plotting facilities could be shared by a number of users or could be made available from drafting suppliers much in the same way as dye-line prints are now. Data would be distributed via floppy disk or telephone and modem.

Software - Drafting

Two-dimensional drafting should in our opinion form the basis of every architectural computing system. Architectural drafting packages as any specialised software must take full account of the particular characteristics of building etc. as well as providing all the usual features normally associated with computerised drawing.

Such features include facilities for the input of lines, arcs, curves and hatching etc; dimensioning and annotating as well as procedures for the combination of such input into components or symbols that can then be stored and recalled later.

In addition any system must include facilities for enhancing the accuracy of input (via grids, traces etc.), editing (of individual lines or components) and for the manipulation of the screen (via pan, zoom, rotate etc.).

The predictable nature of walls, windows etc, combined with the need for different drawings of the same object or building at various scales and levels of detail allow us to add a more specialist feature to our system. It should be possible to input external and internal walls and partitions by simply indicating their endpoints. Such walls could be made up of many layers of either continuous materials (such as plaster, concrete block etc.) or by repetitive components (such as timber studs). Elements such as windows and doors could also contain cavity closing information.

Such input is then interpreted in such a way so as to allow a drawing made at say 1:500 to contain only information about perimeters of external walls. One at 1:100 might represent all walls simply with two parallel lines (which may or may not be hatched), windows and doors being represented by commonly-used symbols. A drawing made at 1:20 could show each layer of construction together with details of DPCs etc. at window and door jambs. Hatching for the various materials is handled automatically and conforms where possible to the existing standards.

Inevitably however, it is not possible to predict all, or even 90% of possible interpretations: so, data produced from such a procedure should be stored (or at least be able to be stored) in a similar manner to any other data entered by the user. Editing should then be available on a line-by-line basis at any level of detail. Insertions could be made without undermining the nature of the construction. (ie. lines inserted in a polygon which represented a wall would be merged into the detail with hatching etc. extended to suit.)

Input is via thumbwheels or small tablet using traditional menu-driven techniques. Current position of the crosshair is continuously updated on the screen, which on the 4114 is implemented by way of a flag attached to the crosshairs which indicates the current position of crosshairs in terms of length and angle relative to previously entered point or temporary origin.

Software - Sketch Design Aids

In considering the possible ways in which a computer could be used during the sketch design process it was felt that neither computer generated perspective nor current analytical routines for daylighting etc. were of any real significance to a designer.

It was however felt that a program which could take freehand sketches and transform them into hardline plans and sections might prove useful in a design studio. Such a program might simply act as a convenient interface between a designer and a traditional drafting package.

Lines freely sketched on a tablet to an approximate scale and represented by continuous streams of x and y co-ordinates may be processed in order to locate nodes, points of inflection etc. This, combined with a carefully designed menu structure, and items such as "current wall type", "current window" etc, could be passed to a drafting package to produce hardline drawings.

Software - Visualisation Techniques

We feel that computer-generated perspectives have a relatively minor role in an architectural computing system and that their usefulness has been much overrated in the past.

The idea of being able to generate 2D plans and sections from 3D building model seems very attractive in theory but proves extremely difficult in practice. If there is to be any connection it should be from 2D to 3D.

Implementation of such a facility should not attempt to model an object in real terms (ie. complete building model) but should rather concentrate on the provision of some basic tools which might be used to model a great variety of shapes.

It should provide the user with sophisticated manipulation routines so that he can position an object easily in 3D space in order to add any graphical enhancement he desires.

It should also provide the user with some sort of programmable interface by which additional modules of software might be added. The ideal language for such an interface might be APL, though hidden-line routines etc. must obviously be written in a compilable language.

Software - Analytical Routines

As with visualisation techniques we do not propose to spend a great deal of time developing sophisticated analytical programs which might interface with our drafting data base directly.

While it is relatively easy to write a program to check condensation across a given construction it would be extremely difficult to get a program to search an entire data base looking for potential cold bridges.

We intend to provide the user with a number of simple calculator-like routines for use at design appraisal stage. Such routines may require additional input of a specialist nature which could have been derived from a date base but would have been far too complex for only marginal benefits. (eg. length and breadth of any given room). Some decisions of this kind about the relationship of analysis to data base have to be made early: others can be postponed until the system is being fine-tuned.

CONCLUSIONS

The time has come for CAD in schools of architecture, and, allowing for the price advantages of new hardware, a suitable means of entry is a drawing system designed as a foundation for a progressively more comprehensive CAD system. CAAD should be promoted not just as its tool but as a unifying force across the whole province of architectural research and teaching. Many of the unresolved problems of CAAD will be better understood from experience of use in a learning rather than a production environment.

ACKNOWLEDGEMENT

The contribution of Michael Costello, of Calligraphic Systems, to the design and development of the system is gratefully acknowledged.