Integrated Computer-Aided Building Design Systems

Dana J. Vanier

Murray W. Grabinsky

Institute for Research in Construction
National Research Council Canada

ABSTRACT

A proliferation of computer aids in the offices of building design professionals has been witnessed in recent years. Many of these aids merely serve to automate the more mundane and time-consuming design office routines. Thus, in an attempt to streamline the design process, efforts have been made, and continue to be made, to produce Integrated Computer-Aided Design (ICAD) systems. However, difficulties in implementing these systems mean that research into ICAD remains as fundamentally necessary today as it was twenty years ago. This paper was produced in an effort to aid those interested in the area of CAD research, specifically software developers and CAD researchers. It provides a summary of existing and prototype ICAD systems and database structures for construction information and shows the evolution of these programs and databases. It also identifies short-comings of some systems and makes recommendations for potential courses of action for researchers and software developers.

Note to Readers

This is considered as a **working document** and not as a final paper. **Comments** from the audience of Working Group 78 are welcome and appreciated. It is hoped the paper in its present form will be of assistance to readers. The paper will be published in due course.

Integrated Computer-Aided Building Design Systems

Introduction

A proliferation of computer aids in the offices of building design professionals has been witnessed in recent years. Many of these merely serve to automate the more mundane and time-consuming design office routines. However, it has long been recognized that the use of a multiplicity of "one-off" applications is inherently inefficient, as the process of transferring and manipulating data from one application to another results in much redundant, and sometimes ambiguous, data being produced, thereby increasing the likelihood of introducing errors. Thus, in an attempt to streamline the design process, efforts have been made, and continue to be made, to produce Integrated Computer-Aided Design (ICAD) systems.

It was found in the course of this research that many of today's (so-called) integrated design aids unfortunately fall far short of the holistic systems envisioned by researchers and practitioners alike. As a result, research into ICAD remains as fundamentally necessary today as it was twenty years ago. It was in an effort to aid those interested in this area of research that this paper was produced.

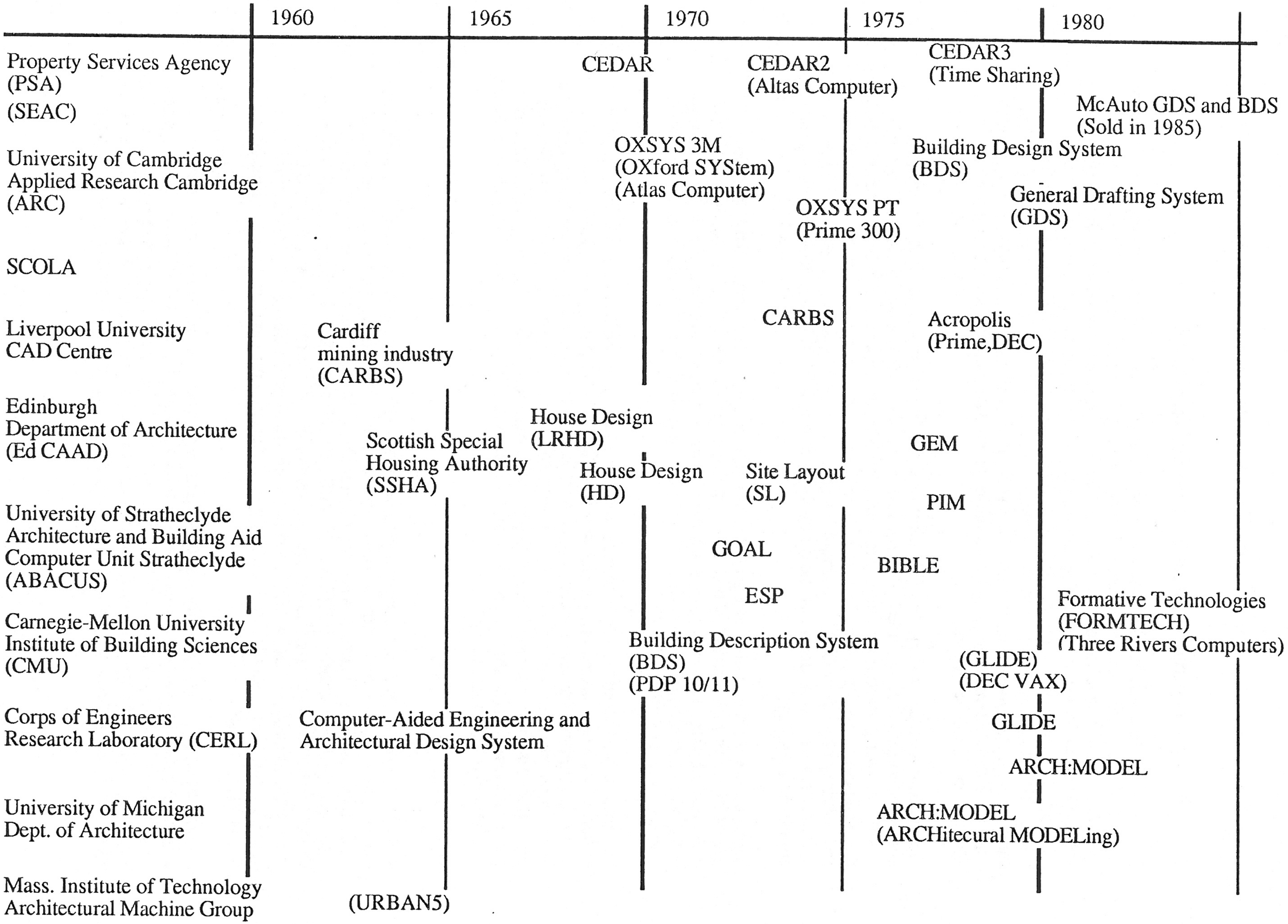
A Report from the USA Building Research Board [Workshop 84] on "an integrated [construction] computer database" has identified the building owner as the main beneficiary of this advanced technology. The development of ICAD systems should therefore meet the building owners' needs. The Board has identified the following major areas for research:

- Analyse data flows in the building process
- Investigate cost savings of ICAD over manual methods
- Ascertain impact of the automation on existing personnel
- Establish appropriate degree of automation
- Encourage support of university research

In the context of this paper, ICAD systems are considered to be computer systems which contain a suite of application programs, each of which access a single, common database. Typically, an operating system is used to control the flow of information between the database and the application programs, ensuring database integrity while allowing many users to concurrently access the data. ICAD must not be confused with computer-aided drafting or computer-aided design drafting (CADD): the research and development in ICAD over the past 20 years has contributed significantly to the development of automated drafting machines that are so commonplace in architecture and engineering offices, but the CADD systems serve only as one application in the integrated CAD system.

This paper will attempt to provide: a clear definition of the problem, critiques of ICAD systems- past and present, an identification of the building industry's need for integrated information systems, and suggestions for schema for building data representation. In this way, the authors will provide an up-to-date review of ICAD for building professionals interested in integrated design systems, to software developers currently involved in the

Integrated Computer-Aided Design Time Line



development of CAD and ICAD software, and to researchers investigating new design tools.

1. Scope of Integrated Computer-Aided Building Design

As mentioned earlier, ICAD systems are considered to be computer systems which contain a suite of application programs each accessing a single, common database. An operating system normally handles the control of the flow of information between the database and the application programs: ensuring the integrity of the database and removing any ambiguity in the data structure. Bijl [Bijl 79] describes ICAD as a unique construction entity, in that it does not exist in the manual world: a descriptive unity providing an image that is comprehensive - containing "one medium" stored memory and, because the form may be three-dimensional, does not contain the redundancies of conventional two-dimensional representation.

Computer-aided design drafting (CADD), at times incorrectly called computer-aided design, should not be included in the scope of ICAD. The form of the data structure for the CADD drawing- normally a sequential vector file of all drawing elements, does not provide the basic elements for integration; namely a strong data structure and a hierarchy based on the building process [Bijl 79]. Although application packages can access CADD data, there is considerable manipulation of the data required to place it in a readable form for the application package.

In the author's definition of integrated CAD, the data base may serve many building disciplines and may extend over the period of design, fabrication and operations. In the first case the information is shared among a number of building disciplines. In the latter, integration can also go beyond the design stage and provide information to constructors, owners, and facility managers.

Integration is, of course, not restricted to the construction professions. Considerable research has been carried out for other design disciplines. Computer-aided design/computer-aided manufacturing (CAD/CAM) is the integration of the design and the manufacturing of mechanical parts. Office tools, such as word processors, spreadsheets and databases, have rearched a high degree of integration in the past two years. Notwith - standing these details, this paper will strictly deal with CAD integration systems developed for the construction industry.

2. History of ICAD

Research in ICAD had its formation at centres of expertise around the world. The authors feel the most significant contributions came from four research groups working with their associated sponsors: Applied Research Cambridge (ARC) affiliated with the Oxford Regional Health Authority (OHRA); South East Architects Collaboration (SEAC) and the Property Services Agency (PSA); the University of Edinburgh's Computer-Aided Architectural Design (EdCAAD) working with the Scottish Special Housing Authority (SSHA); and Carnegie-Mellon University's affiliation with the Construction Engineering Research Laboratory (CERL).

A Time Line (Fig 1) follows the development of prototypical ICAD systems and the evolution to commercial systems. This provides a quick summary of the research centres involved in the development of ICAD products, the evolution of systems developed at these locations, and the computers used for the development of these prototypes or products. An annotated bibliography has been compiled on a large selection of papers, journal articles and books published on ICAD and closely-related topics. This is presently available from the authors and is pending publication.

A working paper, [Spoonamore 85], on a survey of existing ICAD systems was presented to Working Group W78 of the International Building Research Congress (Congress International du Bâtiment, CIB) at their symposium in September 1985. This paper presents a summary of forty-five responses from vendors and developers. Included in the results are type of computer, programming language, terminal manufacturers, input/output devices, three-dimensional capabilities, areas of integration, disciplines encompassed, and component and packaged prices. The survey does not name vendors, but provides general trends and directions for prototypical tools and commercial products. Although an unpublished article, copies may be obtained from the authors or the present chairman of W78.

3. Generic Description Systems

A large portion of the data for this paper was obtained from "Integrated CAAD Systems" [Bijl 79] and Mitchell's work on "Computer-Aided Architectural Design" [Mitchell 77]. Working with this initial information and preliminary research of their own, the authors have tried to "pigeon-hole" the ICAD development types to consolidate the data for the reader.

Each centre of research developed its own description system for the integrated CAD information. Similarities do exist among different systems, but the development depended primarily on the bent of the researchers, the source of funding (i.e client's requirements) and the amount of financing available.

The authors have identified the following breakdown. Integrated CAD information has two major components: alphanumeric information and geometry description. Both have an external (seen by the user) and an internal structure (used by the program); these are defined as logical and physical respectively by Mitchell [Mitchell 79]. However, the overall structure of most ICAD systems is based on the type of construction: component-based or rationalized-traditional [Bijl 79].

The geometry description has strong external and internal structures. The user sees the geometric relationship between parts, this will be based on function, spatial relationship or physical location. The internal data structure of the information is how the computer represents the building and this may be point set, boundary model, or boolean models [Bijl 79].

The same will be true of the alphanumeric information: the internal, or physical, structure will be dependent on the type of database whilst the external, or logical, structure will be based on the relationships of parts or their association with the geometry description.

Component-Based versus Rationalized-Traditional

Systems, such as OXSYS, were developed to represent buildings which used component-based forms of construction. In such systems a building was viewed as a collection of a very large number of discrete components. In this context, a component is considered discrete insofar as the attributes of the component are unaffected by the addition, modification, or removal of neighbouring components. The efficiency of component-based systems relied on the multiple occurence of instances of elements from a well-defined set of building components. An example would be the use of modular components from a catalogue of parts: nothing can be formed or poured. As a result these component-based ICAD systems could only be used from component-based construction, naturally.

Rationalized-Traditional systems, such as the SSHA's House Design, were developed to represent a building as a composition of in-situ, non-discrete building elements. As such, the attributes of an element may be subject to modification as a consequence of a change in a neighbouring element. This information may be computed either interactively or "en bloc", but either way the process is computationally expensive because of the need to describe internal attributes as well as relationships with neighbouring elements. Rationalized-Traditional systems rely on libraries of standard details representing the relationships between adjacent components. An example would be standard construction in North America, a specific detail would be used for the connection of open webb steel joists to the exterior wall of a building. If the type of insulation were to be changed, there would be a necessity to alter the detail.

Geometry Description

To define the different types of geometries it is first necessary to outline the conventional methods CAD researchers have used to describe the physical shape of the building elements. The research investigation has identified two major components of the geometry: the geometry hierarchy (relationship of components) and the geometry representation (internal computer description of shapes). The authors feel, to date, the geometry representation has played too important a role in the development of ICAD and not enough research has taken place to properly develop geometric hierarchies, that is, to investigate what is necessary to create a proper design or maintenance structure. This is identified in case studies [Vanier 85] where the lack of a strong hierarchy, but good representation capacity, made impossible the modification of elements in a complex structure.

Geometry Hierarchy

The authors have selected Bijl's breakdown of the hierarchies: functional, spatial, and physical [Bijl 79].

Functional: Objects are associated according to their functional relationships. That is, a beam is connected to a column and the column to a foundation. It can be general and, as an example, the block, stories, or rooms of a building may be related through their function in the building adminstrative hierarchy.

Spatial: Objects related spatially are identified as sub groups of larger "space volumes". A number of existing systems, such as GLIDE and CEDAR [Bijl 79], can be used to advantage to related these spaces. In the case of CEDAR, two theoretical states of "inclusion" and "exclusion" have been developed to associate volumes. In this example everything is either a part of something else, or it is explicitely not a part of another volume.

Physical: A major difference exists between the physical structures of component versus traditional systems. In component-based, there is a catalogue [Bijl79] of components and instances of theses components. In rationalized-traditional, a strong geometric description of the building procedures is required. It is then "a relatively compact map of elements in a strong relational structure" [Bijl 79]. Objects in both types of systems are associated according to their absolute or relative physical location. This is the weakest form of representation in that only the X, Y, and Z dimension location of the reference points of objects are recorded. This is not a true hierarchy, but a catalogue of locations. Many geometric modellers use this type of representation.

Geometry Representation

The representation of the geometry is an internal matter for the database of the program, but it affects the search, locate, and edit capabilities of each modeller. Simple representation may be "quick and dirty" for simple applications, but may fall apart in rigourous use. Complex representation may be a burden for simple tasks (that is, more difficult than manual representation), but may provide a robust environment for the manipulation of large databases of geometric information. In reality, even the complex representations may not be able to handle, in reasonable times, large data structures. Three types of representation are in current use in the construction industry: point set, boundary model, and boolean model. Others such as octree representation and grid array [Mitchell 77] will not be discussed as no construction ICAD systems presently use these techniques.

Point Set describes a system of nodes and connections. This can be described as a wireframe identifying the nodes (corners) and their relationship (lines, edges) with other nodes. Advantages are that the information is easy to access and to test and the literature is full of algorithms to create and view the shapes. The disadvantage are the large storage requirements for complex geometries and the lack of explicit relationships in the geometric data structure. In addition, the frame work is not good for representing construction information because of need for a multitude of relationships between components, thereby slowing considerably the editing of design data. This is particularily evident for complex shapes where every relationship has to be tested (nodes-connections) before one dimension can be altered. This is the most simple of representation methods.

Boundary Models describe the representation of the surfaces of objects. Surfaces are described with polygon planes and the surfaces are inter-related through the sharing of edges and vertices. Advantages are the compact data storage of the information and the ability to model complex shapes simply. Disadvantages are the complex spatial tasks that must take place to edit information. Shapes must be tested for "well-formedness" [Eastman 80]. This representation appears to be more robust for complex objects than point set, but still requires considerable computation to relate graphics to data. The majority of ICAD systems investigated used this type of representation (GLIDE, GEM)

Boolean models describe solids modelling normally associated with mechanical parts design. Solids are described using shape primitives (cuboids, cylinders, spheroids, etc.) and the union, intersection, and difference of shapes produce the desired model. The methodology of the construct is similar to the creation of the object by milling process which is an additive and subtractive process. The advantage of the system is the "solid" representation of objects and the economical use of data storage space for describing objects. The disadvantage is the high expense in calculation of the shapes. Solid model representation is computationally expensive for large numbers of simple shapes, as is required by the construction industry.

Alphanumeric Data Structure

The external structure of the alphanumeric information depends on a number of factors: the type of ICAD system (component vs traditional), the structure of the geometry description, and the type of internal database. In the authors' opinion this is the area where the least information exists and probably where the most research is required.

The internal data structure at one time was hotly contested; as to what layout provided the optimum performance of the database. The advent of scores of relational and hierarchical databases, along with increased processing speed, disk sizes, and main memory has reduced the importance of this question.

In addition to the alphanumeric information is a whole subset of alphanumeric construction databases created to meet the needs of engineers and architects alike. This is handled in depth in Section 6.

4. Major Thrusts in Research and Development

OXSYS - BDS/GDS - McAuto

Location: Oxford, U.K.
Affiliation: Applied Research Cambridge Ltd.(ARC) (E. Hoskins, P. Richens);
Oxford Regional Health Authority (ORHA)
System: Component-based, 3-D cuboid description, orthogonal
Sources: Bijl, Integrated CAAD Systems.
Mitchell, Computer-Aided Architectural Design.

The Oxford Method of building was developed over a period of some 14 years by the ORHA, and was intended to cover a range of health programme buildings. The method employs a structural steel frame and a well defined set of components based on a modular planning grid.

In 1971 the ORHA, assisted by ARC, undertook the development of a CAAD system based on the Oxford Method. This system, OXSYS-3M, ran on a CADC Atlas computer. The original Oxford Method, 3M, was based on nonuniform planning grid, which was considered to inhibit planning freedom. This, combined with the fact that the Atlas was becoming increasingly unreliable, led to a redevelopment of both the planning method and the computing resources. As a result, in 1975 work began on the OXSYS-PT system which was based on a uniform planning grid and would run on a dedicated minicomputer, the Prime 300.

Building design on the OXSYS-PT takes place in four stages: Brief; Outline; Detailed; and Production. The design process is supported by three levels of software: the Basic Operating System (data management, programming, etc.); a Building Description System (zoning, component description and position, site description, etc.); and a Detailed Design System (design rules of the Oxford Method). All information is stored in three separate databases: a Site Description; a Building Description; and a Codex.

The Codex is a component information file; that is, it contains a physical description of the building components. Since not necessarily every building component will be used in every project, the Codex is divided into two parts: the Master Codex containing all components of the Oxford Method; and the Project Codex containing only the components used in a particular project.

The Codex combined with the Building Description file allows a complete description of the building to be held in computer memory. The database may then be queried according to functional, spatial, and physical relations, and according to component codes and relations.

An attempt was made to make the software as independent of the Oxford Method as possible. In this way the system could find applicability to other building systems as well. The system was still in use as late as 1984, though the emphasis at ORHA had shifted to maintaining and renovating existing buildings as opposed to designing new ones.

The product was marketed by ARC Ltd. as Building Design System (BDS) internationally. A "spin-off" package, General Drafting System (GDS), was well-accepted in the architecture and engineering communities in the 1970's and 1980's. McDonell-Douglas (McAuto) has recently purchased ARC Ltd. and is actively marketing both BDS and GDS in North America and Europe. This is the only early product that has successfully made the migration to the commercial market [Bijl 79].

SSHA

Location:	University of Edinburgh, U.K.
Affiliation:	EdCAAD (A. Bijl), Scottish Special Housing Authority (SSHA).
System:	Rationalized-Traditional, 2.5-D, orthogonal (House Design).
Sources:	Little, The Organisational Implications of CAAD. Mitchell, Computer-Aided Architectural Design. Bijl, Integrated CAAD Systems.

The SSHA was established in 1937 to assist the Commissioner for Special Areas in his task of relieving unemployment in Scotland. The SSHA was particularly active in the post-war period, during which a large portion of its building stock was produced.

Working with EdCAAD, the SSHA first applied computer techniques by rationalizing quantity surveying techniques to automatically produce bills of quantities. Later, a more ambitious program was undertaken, resulting in a rationalization of existing manual methods and these methods being incorporated into the CAAD system. The CAAD system was intended to be independent of the building technique adopted. This was accomplished through the use of standard detail packs, manually produced and maintained. However, attempts to develop a property database from production information predated the availability of adequate technology [Little 84].

The resulting CAAD system was not tied to particular methods of construction, materials, or standard plan types (although a library of standard plans was established), but the system was restricted to one- and two-story houses with level floor plans. The software consisted of two parts: first the detailed design and documentation of housing units and secondly the site layout, appraisal and costing of site plan alternatives.

Originally the CAAD system was intended (and hence developed) for "green-field" sites. However, by the time it was deployed SSHA's workload was facing rapid change, and was now required to renovate the existing building stock and to redevelop exiting sites. Hence, further expenditures on development of the system became difficult to justify.

By 1982, only 10% of the capacity of the main-frame computer acquired in 1974 was being used for CAAD purposes, the remainder being associated with management of the Finance Department. As well, the slackened pace of CAAD development in the late seventies resulted in a number of key personnel leaving the SSHA to pursue developments in other fields and contexts.

BDS - GLIDE - PIM

Location:	Carnegie-Mellon University, PA
Affiliation:	C. Eastman (Inst.of Physical Planning), CERL association
System:	Language for Design Description
Sources:	Bijl, Integrated CAAD Systems. Eastman, Spatial Analysis in Computer-Aided Design

GLIDE, a descendant of BDS, is an interpreted high-level structured programming language. It was intended to assist in the construction of "Design Information Systems", and as such it provided facilities for user-interface, data structuring, procedural processing, and geometric modelling.

A procedural user-interface was incorporated whereby the user could store a series of statements as a batch program for later execution, or could have a statement executed immediately so as to enable interactive work.

As both spatial and non-spatial information were intended to be used, facilities were provided to enable the user to create complex datatypes. These datatypes consisted of Forms (a generic name and a set of attributes containing default values). Copies of the Form could then be made, the values of the attributes of the Copy defaulting to those of the Form's.

The user was able to collect several Items (such as Forms or Copies), and put them into a Set. In addition, several Sets could be collected and put into a Set, enabling the user to construct arbitrary hierarchies. Individual Items could be "put" into or "taken" out of a Set, and procedures could be applied to the entire collection of Items contained within a Set.

Many of GLIDE's geometric modelling features were developed from those of BDS. In GLIDE, a Form could provide a special set of shape attributes for a Copy, allowing for the description of concave, planar superhedron. Hence, the user was able to create parameterized shapes (the topology as defined by the Form, the geometry, or vertex coordinates, specified in the Copy). Provisions were also made to enable spatial set and search operations.

A compact storage structure enabled GLIDE to efficiently describe a large number of relatively simple shapes. However, the shapes were considered to be discrete (i.e. the shape's attributes were not affected by the relation of the shape to its neighbours).

CEDAR - CEDAR2 - CEDAR3

Location:	U.K.
Affiliation:	Property Services Agency (PSA), South East Architects Collaboration (SEAC)
System:	3-D, based on a well-defined Method of Building (MoB)
Sources:	Bijl, Integrated CAAD Systems.

The development of CEDAR (Computer-aided Environmental Design Analysis and Realisation) took place in three phases. The first phase, CEDAR, was carried out from 1969 to 1971 for the PSA, and was intended to investigate the feasibility of developing a CAAD system based on PSA's Method of Building (MoB). Some exploratory programs were written, based on well-defined steel structures, orthogonal and generally single story. The development of CEDAR2, a pilot system, was susequently undertaken. CEDAR2 concentrated on detailed design, and ran on a CADC Atlas 2 at Cambridge. Controlled experiments were begun in 1973, and favourable results led to the development of CEDAR3.

The development of CEDAR3 was necessary for several reasons, including a redefinition of PSA's MoB, and a change in computing resources to a more powerfull time-share computer. Some change in the desired workload led to the development of two phases- 3a and 3b. CEDAR3a concentrated on building, CE, ME and EE capital costs, energy analysis, daylighting and lift selection, while CEDAR3b was intended to cover site description, daylighting with site features, and solar gain.

CEDAR represents a building as a set of "instances" of functional types. An "instance" is composed of rectangular parallelepipeds, and attributes of the instance may be described at various levels of detail. Thus the set of instances form a combined spatial and functional building image. The building description is held in the database in a set of files organized by: functional type, attributes, name, and geometry.

The design process is supported by three levels of software: a Basic Level including the operating system; a Design Level including facilities for inputing the building description, outputing graphics, editing, preparing and controlling and executing applications programs; and an Applications Level which runs the applications programs.

SCOLA

Location:	West Sussex County, Chichester, U.K.
Affiliation:	West Sussex County Council (B. Peters, J. Paterson)
System:	Dimensionally-coordinated, component-based.
Sources:	Mitchell, Computer-Aided Architectural Design. Ray-Jones, Computer Development at West Sussex.

The Second Consortium of Local Authorities (SCOLA), of which West Sussex was a founder, was formed in 1961. At that time a coordinated system of drawings for building components and their assembly was developed. Libraries, schools, and health-care facilities were typical building types for the SCOLA Building program. The system, combined with a method of serial tendering (tendering for a program of work for a period of time as opposed to tendering job by job), formed the basis of a rationalized, dimensionally-coordinated, component-based building system which was developed into a CAAD system.

Designs could be input using a light-pen and a refreshed cathode-ray tube. At the onset of a project an information file, including a list of properties and requirements, was set up. Properties from this file were selected and ranked for each design item (eg. building type, functional group, and material). The computer was then able to aid in the selection of suitable building components to meet performance requirements at each level of design. The software was capable of providing graphic descriptions, cost analysis, environmental and structural evaluations, and automatically generated construction documentation.

Output of the final design was in the form of plotted drawings and automatically-produced bills of quantities. In 1968 West Sussex was also considering the development of a network analysis which would allocate the contractor's resources for him, instead of traditional bills of quantities. Such a development would give direct contact between designer and contractor through the computer.

A change in SCOLA policy led to discontinued use of the system in 1974.

CAEADS

Location:	Construction Engineering Research Laboratory
Affiliation:	Janet Spoonamore, Charles Eastman, Harold Borkin
Sources:	Spoonamore, CAEADS - Computer-Aided Engineering and Architectural Design System

The system was designed to meet many of the needs of USA Military Construction Army. It is composed of a large central design database permitting access to a wide variety of design disciplines: specification writers, planners, architects and engineers.

Modules included in CAEADS are SKETCH, ARCH:MODEL, SEARCH, BLAST, and ABES/CACES. ARCH:MODEL, developed at the University of Michigan, provides full 3-D capabilities to the design packages, including interference checking and transfer of data to other design modules. BLAST is a well-known energy load analysis and simulation program while ABES/CACES prepares cost estimates.

URBAN5

Location:	Boston, Mass.
Affiliation:	MIT (N. Negroponte, L. Groisser).
System:	Research oriented, Interactive.
Sources:	Mitchell, Computer-Aided Architectural Design. Negroponte, The Architecture Machine.

The original objective of URBAN5 was to "study the desirability and feasibility of conversing with a machine about an environmental design project...". To simplify this study, it was decided to commit the system to working under synthetic conditions, and real-world problems were not attempted. For example, the graphic system was abstracted to represent geometry as a collection of 10 foot cubes. Such abstractions distorted some problems, but the resulting simplifications permitted advances that otherwise would have been thwarted in an attempt to provide increased realism.

Although URBAN5 was run on a time-sharing computer it was not used in time-sharing mode, and hence was never fully taxed. This was intentional, as the introduction of future, more powerful, minicomputers was anticipated.

The real advantages of URBAN5 were not in any applications software (for the system was no more than a research toy), but rather URBAN5 investigated the areas of user/computer interaction. It was assumed that the user was completely unfamiliar with the system, and thus URBAN5 would have to "teach its own language, learn through teaching, change from learning, and adapt from changing." [Negroponte 70]. Hence, the user was capable of changing algorithms without actually programming in a computer language, and the software underwent a pseudo evolution, being tailored to a particular user. As a result, URBAN5 suggested true dialogue with an evolutionary and intelligent system- but, in itself, was neither evolutionary nor intelligent.

URBAN5 was eventually abandoned when limitations inherent in its software design, and in the designers' fundamental assumptions about the design process, hindered further development.

GEM

Location:	University of Cambridge
Affiliation:	Ian Braid
System:	Geometric modelling system
Sources:	Bijl, Integrated CAAD Systems.

The GEM system was developed as a geometric modeller for mechanical components such as cast, milled, or turned parts. It is, however, of interest for CAAD systems for two reasons: firstly, GEM was the first system to implement the complete set of spatial operations, and to consider the use of primitives for constructing complex shapes; secondly, the GEM system describes an object as a limited kind of curved polyhedron, demonstrating how systems for planar polyhedra could be extended.

GEM represented an object as a small number of very complex concave, curved zonoids. The user constructed a model by using the spatial set operations applied to a small number of primitive objects. In addition, facilities were provided so that the user was able to create his own primitives. This involved the use of "sweep" operators (a point swept to a line, a line swept into a lamina, a lamina swept into a volume), volumes of revolution, etc.

The final shape was *not* stored as a sequence of primitives and operations. Instead, the final shape description was stored as a topology and geometry in a subset of Baumgart's Winged Edge structure (edges linked to a face, vertices linked to an edge, vertex coordinates, and face equations).

Importantly, the system was intended purely for *shape design*, and contained no facilities for associated attribute data.

ICES - STRUDL

Location:	MIT, Mass.
Affiliation:	D. Roos *et al.*
System:	Integrated Civil Engineering Systems
Sources:	Mitchell, Computer-Aided Architectural Design

ICES (Integrated Civil Engineering Systems) was developed at MIT in the mid-1960's. It contained an architectural subsystem call BUILD and a structural analysis and design component called STRUDL (Structural Design Language). Internal to ICES was a programming language called ICETRAN that was a superset of FORTRAN. This permitted accessing the database and creating programs to read the proper information and to output the results in the proper form. In addition there was a Command Definition Language (CDL), provided for defining the external form of ICETRAN.

There is still considerable interest in both ICES and STRUDEL, mostly in civil engineering.

AD 2000

Affiliation:	Patrick Hanratty
System:	AD 2000

This system was mentioned owing to its impact on the CAD drafting field over the past 20 years. ADAM 3, developed in the early 70's, became the father of a large number of CAD drafting systems. This was primarily because of the system design: maintain software and input/output independence. Because it was one of the first CAD software packages and because of its transportability, it was quickly ported to a wide variety of machines.

AD 2000 is a 3-D system that was not restricted to only 3-D. It has strong capabilities in mechanical drafting and numerical control, but has limited capabilities in architectural and building engineering. There were many claims to "design" capabilites and these were incorporated in later versions of the software. It was adopted by Los Alamos Scientific Laboratory as an "organization standard" and was successfully implemented on 3 different mainframes and 3 different minicomputer systems. This lead to the wide acceptance of the software by "turnkey" manufacturers.

ARK-2

Sources: Mitchell, Computer-Aided Architectural Design.

ARK-2's claim to fame lies in the fact that it was the first commercially-marketed integrated CAD package in the United States. At its conception in the mid 1970's the main computer was a 16 K machine and the entire system, including software, was sold for $160 000 (US). It was using an old PDP mini-computer and had the facility to store both project databases and library information, as well as, performing standard application programs [Mitchell 77].

5. Advantages and Disadvantages of Specific Systems and Ideologies

General Features

Many of the systems developed in the 70's were client-based and involved extensively, if not completely, researchers from the academic community. Many of the common problems which plagued these systems were due to deficiencies in the technology at the time. The most obvious examples are the use of cathode ray tubes, LSI computers, floppy disk systems, and programming languages.

Cathode ray tubes used vector drawing as the form of information output. Color and interactive motion were difficult, if not impossible. This implied that the standard for graphical presentation was point set representation and surface modelling was not possible until the advent of raster screens. The user interface at the time was the Tektronix 4014, the workhorse of the CAD industry, thereby forcing the user to use pickmenus and thumbwheels. The evolution of interface design with the advent of Macintosh technology has opened many doors to software developers.

Large Scale Integration (LSI) computers limited the main memory size to 64K or 256K. This permitted some applications for small data sets but it was much too small for the large application of Integrated Computer-Aided Design.

Floppy disks limited file storage space, although prohibitively expensive hard disks were available at research laboratories. Disk sizes of the order of 10 to 20 megabyte range were maximum for mini-computers. These are now a standard for personal computers.

The majority of systems were implemented in FORTRAN, the engineering standard even today. This prohibited the development of modular software and greatly reduced the I/O possibilities in the early CAD products.

The problem of efficiently and completely representing building geometry was, and remains, a difficult problem. The problem was resolved in the early systems by imposing certain restrictions thought to be "livable", if not desirable. These restrictions included 2.5–D, orthogonality, and convex shapes.

Some Characteristics of Specific Systems:

OXSYS:

- Covered health program buildings
- 3-D, Component based
- Creation and comparison of design alternatives in early stages

- Early cost and environmental analysis
- Phases of Work: brief, outline, detail, and production
- Box Geometry for simplification, but restrained to orthoganal
- Strong hierarchies of building components and of spatial and functional zones
- Use of CODEX for standard elements; PROJECT CODEX as a subset
- Representation of spatial (overlap, enclosure, adjacency) and functional (zoning) relations
- Exhaustive search through trees (forward chaining)
- Limited Building types
- 500 bytes per metre square
- Output as plotted production drawings and schedules by location and zone

House Design and Site Layout (SSHA):

- Intended for 1- or 2-story house designs with level floorplans
- 2.5D, Rationalized-Traditional system
- It recognized objects as being non-discrete, and recognized the relationships between construction components.
- Used manually created and maintained "standard-detail packs"
- Permitted the creation of a standard plan library
- Intended for "green-field" studies, no rehabilitiation capabilities
- Detailed design and documentation facilities provided

BDS-GLIDE:

- An interpreted high-level programming language intended for developing Design Information Systems
- Datastructuring capabilities allow user to create arbitrary hierarchies and apply default information
- Geometry/Topology/ and relationships embodied in the building descriptions.
- Large number of simple shapes with the possibility of complex shapes.
- Form and copy of form permitted the cloning of information for the easy input and the data verification process. Identification of copies of form are mandatory, thereby being able to identify items that are almost the same as.
- Provided for the fast spatial access, a necessary feature for building design systems.

CEDAR:

- Public building programme (Post Offices, Telephone Centers,..)
- 3-D, based on a well defined method of building (MoB)
- A doodling process allowed rapid comparison of design alternatives; no overall design sequence was imposed
- Early cost comparisons, energy analysis, daylighting,...
- Standard defaults could be employed
- A LOOK'N'CHANGE mode allowed viewing alternatives without changing the database
- Building geometry represented as paraxial parallelepipeds (the max/min x,y,z coordinates)
- Spatial relations are stored explicitly; functional relations are derived
- Define BLOCKS; assemble blocks to form a BUILDING; define STORIES within blocks; define SPACES within stories.

CAEADS:

- Developed to aid the design of buildings in Military Installations
- Specifically developed for one large client

URBAN5:

- A research tool used to study user interaction with a computer about Urban Design problems; never intended for implementation
- Abstractions used to simplify the system (eg.the world was approximated with 10' cubes)
- It was assumed that the user was unfamiliar with the computer; therefore the system had to be self teaching and had to be able to "learn", evetually being tailored to the individual user

GEM:

- A geometric modeller for mechanical components
- Represented an object as a few very complex shapes
- Final shape description stored as a subset of Baumgart's Winged Edge datastructure
- Non-spatial data not included in the datastructure

ICES - STRUDL:

- Common operations modules
- Modular software design

AD 2000:

- Vector generation of geometry is restrictive.
- Need to tie data to geometry, or better yet, the geometry to the data.
- Minimal machine dependency
- Data must be independent of computer architecture
- Growth possibilities must be possible
- Input/output must be device-independent
- Provide exhaustive geometry capabilities
- Heavy use of COMMON Block

6. Construction Data Information Formats

To propose a system to meet the requirements of the building industry it is first necessary to identify existing construction information data structures and thesauri. These have been in existence for a number of years and originated because of the need to control the vast amounts of data in the construction industry. There have been a large number of construction industry classification systems developed over the years. The majority are extensions of the Universal Decimal Classification, the predominant library classification method in Europe. Examples of these are the Abridged Building Classification (ABC) [Giertz 81] and the Extended Building Classification for Architects Builders and Civil-Engineers (EBC) [Fink 69], providing an extensions to the 62 classification for engineering works, 69 for buildings, and 72 for architecture of the UDC. Although they are excellent for cataloguing technical information, they are restrictive for cataloguing building components because of the limited number definition, the lack of location information attributes, and the lengthy form of the annotation (eg. 728.2.011.263 - Three storey residental building).

SfB Basic Tables:

History:

The SfB (Samarbetskommittén för Byggnadsfrågor) [Giertz 82] was developed in Sweden in late 1940's. The original uses were in construction specifications. In Britain, in the 50's and 60's the Co-ordinated Building Communication system (CBC) was developed as a computer-operated management and costing system based on the Sewish SfB.

Description:

There are two major components of the identifier codes: the building elements and the construction method. The intention of the coding method is to simplify information for the industry, not to make it more complex. It was decided that construction methods needed classification without reference to the elements produced. The coding method recommended the use of the Universal Decimal Classification (UDC) to augment the SfB for product literature.

It is broken into Elements, Construction form, Labour and Plant, Materials, and Location Codes.These could be placed in any order by the user depending on the needs (Element - Construction Form - Labour and Plant):

> Elements are function-based and include: 20 - superstructure, 50 - site services, and 43 - Site Finishes: Floors.
>
> Construction forms are material based and provide a descriptor for the type of installation: G - Structural Units, M - Foldable Sheets, and V - Thin Coatings.
>
> Labour and Plant identifies the responsiblities and or the materials to be used: "a" through "d" describes administration though operations, and the remaining lower case letters identify specific materials such as natural stone (e), loose fill (p), and paint materials (v).
>
> In total this produces an identifier (43) Vv for "painting of floors".
>
> Location codes are project-specific and identify the location of materials, details, or procedures. It is a five-tiered hierarchy identifying the sectors of a building (1 to 9), the blocks (00 to 99), the storey (00 to 99), the departments (00 to 99), and the rooms (000 to 999). In total this produces a 10-number identifer code for the location (eg. 1 02 10 14 007 for room 7 in department 14 on the 10th floor of the 2nd block in the building).

The advantage is that it is more of a project system than a technical information filing system. The disadvantages are that it has been superceeded in the U.K. by CI/SfB. In 1966 65% of architects, 55% of quantity surveys and 45% of contractors in the UK were using the SfB.

CI/SfB:

History:

The Construction Index for Normalization of Building Information [Ray–Jones 76] was developed in Sweden in the mid-50's to address the needs of cataloguing construction industry information, reports, drawings and slides. The work was initiated through the CIB (the International Council for Building Research Studies and Documentation) as a joint development project to adopt the Swedish system of cataloguing information - SfB. In the beginning 52 systems were evaluated and two were identified as being the most useful: UDC - Universal Decimal System and SfB. CI was appended to the SfB to identify the U.K. version while SI and BRD are affixed to the respective French and German versions.

Description:

The system employs a multi-table layout of descriptors for the identification of reference materials:

Table 0 - identifies the physical environment [e.g 522 defines Entertainment Facilities (51) specifically Music Halls (__2)]
Table 1 - identifies the elements [e.g. 21.1 defines Structure Primary (2_) specifically Walls, external walls (21) and more detailed the Load Bearing walls (__.1)]
Table 2 - identifies the construction, forms [e.g. F defines Blockwork]
Table 3 - identifies the materials [e.g. f4 defines lightweight Cellular concrete]
Table 4 - identifies the activities [e.g. F5 defines the Shape (F_) and the Size attributes (_5)]

The descriptor 522 21.1 Ff4 F5 refers to the size of lightweight cellular block for loadbearing walls in music halls. Boolean operations may be used to further describe specific information, so the descriptors 33/35 or 33 + 37 or 33:4 mean "through", "and", "controlling" respectively. This is a variation from the SfB, enabling the user to specify Block, Storey, Department and Room as location identifers.

The advantages of CI/SfB are that it is well-known internationally and it has sufficient history that it is accepted internationally as a documentation method. It also can be used effectively in the project management field as it identifies components and can provide locations. There is little duplication of information and retrieval of information is greatly enhanced. It can be used on drawings for the co-ordination of details, specification for the reference of materials, and co-ordination of drawing and specifications.

The disadvantages are that it is difficult to use, there is some redundancy of information, and all information has to be filed at all times. It also increases the burden on the designer by constantly requiring the addition of extra information for most components on drawings and specifications.

Masterformat (16-Part Divisions)

History:

Known as the 16-Part Division specification, the Construction Specifications Institute (CSI) in the United States developed the "CSI format for Construction Specifications" in 1963 subsequently changed to the Uniform System in 1966 [Masterformat 83]. In Canada, the Building Construction Index (BCI) was compiled in 1966. These merged in 1972 to form the Uniform Construction Index (UCI) and was renamed the MASTERFORMAT in 1978. It is available as MP -2-1 from the CSI and as Document 004E from Construction Specifications Canada. Since that time the National Master Specification in Canada and the SPECTEXT in the United States have used the MASTERFORMAT as the general basis for its titling and numbering system.

Description:

In addition to dealing with the bidding requirements, contract forms and conditions of the contract, the MASTERFORMAT outlines the 16–divisions for the construction specification. It is organized on a 5-digit numbering system, the first two identifying the construction general category of work (e.g. division 02 - sitework, division 10 - specialities) and the other three digits identifying a specific construction element or system (e.g. landscaping - 02900, folding partitions - 10650).

The advantages of MASTERFORMAT are that it is a well-organized system used across the construction industry in North America. It assist in the writing of specifications and cost estimating. It has a degree of flexibility that permits the specification writer to add his own level of detail. It is closely tied to construction trades organization and tendering practices in North America. It can be used as a product technical data file system. The disadvantages are thaqt it has a limited numbering system and the system is trades related. It cannot deal with specific detailing for elements, and finally, it is procedure-based, not component-based, thereby restricting its use in complex data structures.

BEAM - Construction Information System

History:

Beam had its beginning at Industry Trade and Commerce Canada in approximately 1972. The intention of the program was to increase the efficiency and productivity of the building process [Holmes 70]. An extensive survey of user needs identified a need for better, quicker to access construction information. These needs pertain to products, codes, technolgy, and commercial data.

Description:

The BEAM system developed a thesaurus or common construction terms language, it developed a standard format for the presentation of information, and it proposed access, store and retrieval methods.

Usage:

The BEAM program was discontinued in the mid 1970's.

I.F. Thesaurus

History:

The I.F. Thesaurus, developed for the Department of Industry, Trade and Commerce of the Government of Canada by the IF Research Group, University of Montreal, was formed by merging the BEAM Thesaurus with the IF Thesaurus and then completing the logic structure [Thesaurus].

Description:

Information procurement is important to the construction industry and to construction sciences. However, finding information in a particular area of interest may be difficult as documents may deal with more than one subject, the document may not always be classified according to all the information it contains, and fields of interest are constantly shifting making it difficult for conventional classification systems (eg. U.D.C., SfB,...) to keep pace. Thus, the search for information may be obscured by the overwhelming amount of information available.

The thesaurus is intended to be a tool utilized to control the terminology used when the natural language of documents, indexers and users is transposed into a stricter language for documentation handling and information science. It is both a guide to the selection of adequate descriptors during the indexing of documents and an aid to the selection of appropriate key-words during the formulation of a request from a user. It is based on the principles of *concept coordination*, and hence it enables all concerned in the construction industry to use the same words to express the same concepts. Although the Thesaurus principally deals with construction and construction science, other disciplines such as physical sciences, human sciences, applied sciences, and earth sciences have entered it's domain. The T-C/C-S Thesaurus includes two distinct parts which have complimentary roles in use:

1. The Alpha-Hierarchical Listing, which comprises a collection of main entries arranged alphabetically. Each main entry is associated with a group of terms describing hierarchy level, related terms, and short notes.
2. The Alpha-Permuted Listing, which includes permuted forms of terms found in (1.). This enables the user to locate descriptors rapidly by reference to any of the words out of which they are composed.

It permits both the indexers of documents and the users of those documents to use the same words to express the same concepts, thereby facilitating easier procurement of information. Another advantage is that the T-C/C-S Thesaurus is the largest part of a conceptual *Mega-thesaurus*, which ensures compatibility between various thesauri being developed by the IF Team.

A disadvantage is that effective usage is dependent on the principles of indexing and document handling adopted by each particular agency.

7. Technological Advancements: Past 5 years to Next 5 Years

Many man-years of work are placed in the development of any large integrated system. It has been proven empirically that flexibility and device-independence will extend the life of existing programs. The future holds much in store for ICAD with both hardware and software innovations.

Hardware considerations include RAMMABLE programs, large disk storage, personal work stations, speciality graphics processors, high speed networks. The evolution of this technology will permit centralization of data, will speed-up processing of information, will permit better integration of software, and will increase I/O speeds.

Software innovations include parallel processing, new user interfaces, object-oriented graphics, transportability, evolution of new languages, data base management systems evolution, standardized operating systems. All of these directly improve ICAD systems.

8. General Conclusions of Advantages and Disadvantages

Bijl [Bijl 79] stated that current (1979) ICAD systems "must be seen as the first steps in a development" and has listed his observation of systems at the time. He identified the need for "arbitrary planar 3–D geometry" over "orthogonal", research to develop software to handle large number of interrelated polyhedra, and high "man-machine interaction" on newer technology equipment. He was careful to identify the need for new techniques in computer sciences and software development permitting the easy maintenance and modification of programs and allowing the user to move one step farther away from the data structure.

The research in this field has clearly indicated the "drop-off" in the years from 1980 to 1985 [Grabinsky 86]. This could be due in part to funding restrictions at a number of federal government department, to the realization that the systems were not performing as envisioned, to the severe hardware restriction at the time, or to the migration of research(ers) to other channels, such as commercial computer-aided drafting systems or knowledge-based expert systems.

The Building Research Board [Workshop 84] has identified that integrated CAD tools would reduce design time, thereby allowing these funds to be rechannelled in design. These tools would also provide proper information for building owners- providing lower life cycle costs with the possibility of passing these savings to tenants.

Many of the following points appear to be comon sense, but still not all ICAD packages include these concepts in their system design:

- Integrate system from concept design stage to facilities management operations
- Address the needs of one client, do not provide ubiquitous solution
- Use internationally-known coding systems such as MASTERFORMAT or CI/SfB
- Provide a frame of information that does not require all the information to function, but can be augmented when data is obtained.
- Pay strict attention to the user interface design.
- Represent both spatial (overlap, enclosure, adjacency) and functional (zoning) relations
- Allow users to create arbitrary hierarchies and apply default information
- Permit the cloning of information for the easy input and the data verification process through "form" and "copy of form"
- Box Geometry is adequate for construction industry, but provide possibility of complex shapes
- Provide exhaustive geometry capabilities
- Provide an interpreted high-level programming language for developing design information systems
- Data must be independent of computer architecture
- Input/output must be device-independent
- Employ modular software design
- Permit the creation of a standard plan library
- Create default files and "expert systems files"

References

[Bijl 79] Bijl, Aart, Integrated CAAD Systems, EdCAAD Studies, DoE Project DGR 470/12, Edinburgh, 1979

[Eastman 77] Eastman, Charles, Spatial Analysis in Computer-Aided Design, Applied Science Publishers, Essex, England, 1977

[Eastman 80] Eastman, Charles, Prototype Integrated Building Model, CAD Journal, vol.12, no. 3, May 1980

[Fink 69] Fink Daniel, Extended Building Classification for Architects Builders and Civil-Engineers (EBC), Danish National Cantre for Building Documentation, 1969

[Giertz 81] Giertz, Lassé M., Hughes, Nöel J., Abridged Building Classification (ABC), An Foras Forbartha, St. Martin's House, Dublin, 1981

[Giertz 82] Giertz, Lassé M., SfB and its Development, An Foras Forbartha, St. Martin's House, Dublin, 1982

[Grabinsky 86] Grabinsky, Murray W., An Annotated Bibliography on Integrated Computer-Aided Design, {available from DJ Vanier}

[Holmes 70] Holmes, B.W., Beam Program - Construction information System, Department of Industry, Trade and Commerce Canada

[Little 84] Little, S., The Organisational Implications of CAAD, CAD 84, Brighton, Apr 3-5, 1984

[Masterformat 83] Masterformat - Master List of Section Titles and Numbers, Naval Publications and Forms Center, Alexandria VA, 1983

[Mitchell 77] Mitchell, William J., Computer-Aided Architectural Design, Van Nostrand Reinhold Company, New York, 1977

[Negroponte 70] Negroponte, N., The Architecture Machine, Cambridge, Mass., 1970

[Ray-Jones 76] Ray-Jones, Alan, Construction Indexing Manual, RIBA Publications Limited, London, 1976

[Spoonamore 82] Spoonamore, Janet H., CAEADS - Computer-Aided Engineering and Architectural Design System, Technical Manuscript P-133, CERL, United States Corps of Engineers, Aug 1982

[Spoonamore 85] Spoonamore, Janet H., Vanier, Dana J., Christiansson, Per, A Survey of Integrated CAD System, CIB W78, Rotterdam, 1985

[Thesaurus] Canadian Thesaurus of Construction Science and Technology, Industry Trade and Commerce Canada

[Vanier 85] Vanier, Dana J., Three-Dimensional Visualization: A Case Study, CAAD Futures, Delft, 1985

[Workshop 84] Building research Board, The 1984 Workshop on Advanced Technologies for Building Design and Engineering, National Academy Press, Washington, 1984

[Workshop 83] Building research Board, A Report from the Workshop on Advanced Technologies for Building Design and Engineering, National Academy Press (Aug 83), Washington, 1984

Author's Address for Comments and Copies

Dana J. Vanier
Institute for Research in Construction
National Research Council Canada
Building M-24
Montreal Road
Ottawa, Ontario CANADA
K1A 0R6

Tel (613) 993-9699