A COMPUTING ENVIRONMENT FOR THE APPLICATION OF SYSTEM BUILDING IN ARCHITECTURE :

A Computer Based Building System within a General Computing Environment for Architectural Practice

by

Stefanos A. Kalligas

Diploma of Architecture,
National Technical University of Athens,
Athens, Greece, 1983

SUBMITTED TO THE DEPARTMENT OF
ARCHITECTURE IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN ARCHITECTURE STUDIES

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1986

Signature of Author ____________________
Stefanos Kalligas
Department of Architecture
January 17, 1986

Certified by ____________________
Patrick Purcell
Associate Professor of Computer Graphics
Thesis Supervisor

Accepted by ____________________
Julian Beinart
Chairman
Departmental Commitee for Graduate Students

A COMPUTING ENVIRONMENT FOR THE APPLICATION OF SYSTEM BUILDING IN ARCHITECTURE :

A Computer Based Building System within a General Computing Environment for Architectural Practice.

by
Stefanos A. Kalligas

Submitted to the department of Architecture on January 17, 1986
in partial fulfillment of the requirements for the Degree of
Master of Science in Architecture Studies

ABSTRACT

Complexity, size and sophistication of buildings, available materials and resulting solutions for their assembly, as well as our needs and expectations for flexibility and optimal performance of the built environment and conservation of valuable resources, constantly increase, and are likely to continue to do so in the future.

This document is based on the idea, shared by several individuals over the past three decades, that if properly utilized, the systems approach to building can for the greater part of the built environment lead to solutions which are closer to meeting the personal, social and cultural needs of the users.

This thesis however, is also based on the assumption that in order for the full potential of systems building to be utilized it must become particularly easy to incorporate building systems and subsystems, selected from the open market, in the majority of the various building projects. If this can be easily achieved, regardless of the size of the projects or the size of the firm that is commissioned to design them, it is believed that Systems Building shall be disassociated from the production of a sterile and mass-produced type of built environment. At the same time its beneficial influences related to speed, efficiency, quality and sophistication of design, will be more generically felt and appreciated.

Substantial reorganizing of the building process is required so as to allow design firms, users of the various building systems, to have easy and affordable access to the increasingly large amounts of relevant information. This enterprise, can be made possible today by the introduction of the digital computer.

This process is facilitated by the fact that computers are already gaining rapidly ground in architectural firms, utilized primarily in the performance of several repetitive tasks. Earlier developments in the area of Architectural Computing are combined today with recent research in the area of expert systems and artificial intelligence. These developments combined with a significant decrease of the cost of the hardware of computer systems, favor the utilization of such systems in the fragmented building industry and the various architectural firms in particular.

The application of the systems approach to building provides the theoretical foundation for the required changes in the building process in order for increasing needs to be met. The introduction of the digital computer provides the technological breakthrough which will enable the required efficient manipulation of extensive information. Systems building and computer developments in several ways complement each other, and from their interface and parallel appropriate utilization, the building process can be considerably influenced for the better.

This thesis introduces the notion of a Computer Based Building System, which is the new improved product of the evolution of the conventional paper based building system so as to meet the new standards in the manipulation and transfer of information. It also focuses on an appropriate Open Computer System for Architectural Practice prerequisite for the utilization of a C.B.B.S. This computer system is open, modular and integrated, and expands according to the needs of the firm. It can therefore easily accommodate various C.B.B.S. depending on the needs of the individual projects without particular effort from the project architect. The parallel utilization of the Computer Based Building System and the Open Computer System for Architectural Practice, provide an appropriate computing environment for the application of Systems Building in Architecture.

The introduction to this document includes a personal statement by the author which identifies the rationale and the ethos for presenting a Computer Based Building System approach to designing in the built environment. The core of the text is divided into three major sections. Section one, contains background information on Systems Building and Computer Applications in Architecture and the Building Process. Section two, introduces the notions of the Computer Based Building System and the Open Computer System for architectural Practice, components of the proposed appropriate computing environment for the application of Systems Building in Architecture. Section three, describes computer programs, components of the C.B.B.S. and the O.C.S.A.P., developed over the past 18 months which illustrate some of the principles presented in this document.

Thesis Supervisor: Patrick Purcell
Title: Associate Professor of Computer Graphics

To those few beloved individuals
who were a constant source of inspiration
throughout this study

ACKNOWLEDGEMENTS.

I wish to thank my advisor, Professor Patrick Purcell for his guidance throughout this work. Through my work with him over the last two years I came to appreciate him both as a teacher and as a person. I owe to him my initial interest in computer applications in architecture, a very valuable teaching experience which I gained assisting him in his course, as well as a lot of inspiring interaction later during my research.

I would also like to thank Professor Eric Dluhosch for his guidance to my studies and his help in my career. His deep knowledge in the area of systems building has been critical for the first stage of my studies at MIT and has served as the basis for my future explorations.

I wish to express my gratitude to professor Joseph Ferreira director of the Computer resource Laboratory of the department of Architecture and Planning, for providing me with the opportunity to perform inspiring research work for the Project Athena over the past 18 months and for being a very agreeable and caring employer. Under his guidance the Computer Resource Laboratory has been a very stimulating and friendly environment for work and research.

Above all I wish to thank my family and good friends back home with whom I share common values and ideals. My work has constantly been inspired by the desire to find some links between the glorious past and a promissing future, for the benefit of my home country. I feel very grateful to my parents for providing me with these incentives and sense of belonging which I find to be a very precious gift.

Finally I wish to thank my good friends and coleagues for their support and advice, as well as the good times we've spent together. Bizhan Azad, for being the nicest "mediterranean" guy in the lab, always friendly and ready to help, initially a "good boss" and finally a good friend. Steven Ervin, for lots of useful advice and sharp comments and Mark Gross For being the wisest "guru" of us all. Joy Hecht, for being a very good friend and when it comes to paste-ups a very efficient one too. Peter Jurgensen, for being a very good and efficient programmer specialist in quickly adapting text formatters for the needs of an individual thesis. George Karniadakis, for being a good room mate always available for late night philosophical conversations and Costantine Kriezis for his love of Greece and his good heart. Christian Marion, for being a very inspiring French philosopher, Ben Olasov, for late night mushroom searching at Wellesley and Therese Vien for some surprisingly good chinese cooking.

I am especially grateful to Aspassia for the patience she has shown over the last year regarding these long sessions at the computer. I thank her sincerely for her love and care.

TABLE OF CONTENTS.

INTRODUCTION

INTRODUCTION

The building industry, when compared with other production industries, is unique in many ways, particularly in the separation of the design function from the production facility and in its marked fragmentation into various professions, trades, and manucturing agencies. Among them, architects are responsible for making suggestions within a given set of constraints, for the development or rearrangement of the built environment, in order to make it more appropriate to suit the users' needs.

However the intellectual process which is required for this to be efficiently achieved, which we commonly tend to refer to as design, cannot be automated, is very difficult to optimize and has no apparent shortcuts, facts which make it very inspiring and challenging for the individual that performs it, but at the same time very expensive for the one who receives the services. This cost tends to be further increased, the better and more meticulously the design process is performed.

Furthermore the design services are still far more expensive to the end user in buildings than in any other products which are industrially produced.

This can be justified to a large extent by the fact that in the building industry the design product, i.e. the building, is in permanent relation to the ground. This fact prevents standardization beyond a certain degree, and limits it to elements of a certain scale. Furthermore the permanence of the structure introduces the need for some organic relation with the ground. Both these constraints do not allow the mass design and production of buildings like other industrial products.

Thus the final product of the design process, the building, is actually a prototype which will never be realized, i.e. industrially mass-produced, and supplied to the users at a low cost due to economies of scale.

This implies that although the user is actually provided with a product that is in several ways deficient as all prototypes tend to be, he has to be subjected to increased expenditures for the design services that have lead to the production of this prototype. Since the end product is a single building the cost of the design is bound to be only a fraction of the cost on the final product while in other industries the cost of the design of an industrial product by far exceeded the cost of every produced item that reaches the market and the end users. Therefore although a building is for this reason likely to be less efficiently designed than a car or an aeroplane, the user has paid a larger percentage of its cost for this "inefficient "design.

These are some of the inherent difficulties in the efficient design of buildings. Architectural design performed in the conventional way, can produce very stimulating results and can lead to the creation of very agreeable environments, however it is a very expensive process which can only be applicable to a small portion of the built environment.

Architects both in the past and today, like other professional groups tend at large to provide their services to those who can afford to pay for them and thus are practically limiting the scope of their work to large projects, either commercial or public buildings, or smaller buildings for relatively wealthy individuals who have the means and the will to hire them.

Indeed it is a very stimulating experience to be involved in the design of a project of such a magnitude or social importance, that justifies sound research and a careful and meticulous design process which is likely to pay off and be appreciated. However, this document has not attempted to deal with this small portion of the built environment which is currently being produced, as it always had in the past, with quite satisfying results due to the direct relationship of the architect with the client, and in some favorable cases has contained architectural creations of the highest value per se which have been so much praised throughout time. Instead this work focuses on that larger portion of the built environment that currently isn't and cannot be designed with the application of conventional architectural services due to reasons previously discussed.

Historically, this environment was developed without the consultation of a professional designer. In relatively static societies, the user either by himself or with the aid of craftsmen build functional structures of small scale which satisfied immediate needs. Decisions were based on a very clear understanding of these needs, as well as a very clear understanding of available materials and techniques, relying on tradition and long evolving primitives which were to be repeated or adjusted according to the specific requirements of the particular job. This context was responsible for the production of what we now call "folk" architecture which is so much admired today for its aesthetic qualities and to which we so often turn for lessons on the apparent hierarchy in the organization of space, in the organic relation with the natural surroundings, as well as the significance of time in the design and development of the built environment.

In modern times, in occasions where tradition still survives and there is an uninterrupted culture within a particular environment which dictates a system for building (both materials and process), a common set of values defining need and standards of measuring their satisfaction, as well a common language for the communication among the various participants in the process, things work out pretty well and the results are pleasing both to the users and to the observer. This, in other words is the situation where a non designed building system exists evolved from tradition. Examples of such an setup can be found in the various "vernacular" environments in several parts of the world and specifically in the U.S.A. in the stick-built, Balloon frame or platform frame house which is the product of continuous evolution and gradual industrialization of a building system for more than 200 years.

Unfortunately the occasions where such situations exist are relatively rare, particularly in the western world. In most cases rapid economic development, a discontinuity in culture due to rapid social change, as well as introduction of new needs and the resulting technological breakthroughs and new materials and building techniques in order to satisfy them, caused a great degree of confusion which is vividly reflected in the built environment.

Indeed with the patterns of living rapidly changing to adapt to a much faster pace with needs for much more efficiency and increased productivity, our expectations from the built environment have too been raised. We require more and more complex buildings with increased expectations for optimal environmental performance, as well as flexibility in order for them to be able to adapt to our constantly changing needs. Furthermore we have a much larger variety of materials and building techniques to select from and the designer is overloaded with relevant information which he often finds himself unable to handle efficiently.

More so the process of making optimal design decisions in order to satisfy these constantly increasing expectations in an environment which is constantly changing becomes increasingly difficult, does not allow for the efficient utilization of precedent and past experience, and for several cases can be described as an almost arbitrary process.

The considerable cost of this process is rather unfortunate for the majority of the built environment since it leads either to oversimplification of the design process with obvious consequences or alternatively to adoption of scenarios which totally exclude the user from the decision making process.

Such strategies are responsible to a large extent for the rather poor degree of satisfaction of users' needs form their immediate built environment which characterized the majority of the built environment produced today. There is indeed an urgent need to increase the efficiency of the building process, which will result in a better satisfaction of these needs.

Two alternative methods have been proposed which tackle this urgent issue in a different way. One is participatory approaches and the other is systems building. Extensive literature exists on both as well as in ways they can be efficiently combined towards a optimal result.

Participatory approaches advocate that the solution to the problem is to involve the end user in the decision making process. The user has his own defined level of action within which he has freedom to act and impose changes on the built environment that surrounds him. This leaves the designer relieved from the burden of having to make the decisions himself, and the user pleased with his immediate environment which suits his needs better since he himself has made the decisions according to his individual needs. Participatory approaches can be particularly helpful in some contexts, especially for housing applications.

However the modern world and especially the western world does not always favor such approaches, since it does not tolerate the resulting waste of resources which is necessarily implied by the fact that an individual is directly responsible for the creation of his immediate built environment. This is more so in cases where increased mobility of the population due to cultural or working habits, does not permit such a luxury, or some emergency urges the fast erection of a large amount of buildings.

In these cases Systems Building is the only strategy that if efficiently utilized can provide an answer. The goal is to increase efficiency in the building process i.e. conserve less resources in the creation or modification of the built environment and produce a result which more accurately reflects the needs of its users.

This is achieved by standardizing smaller sections of the buildings, building subsystems or functional units and industrially producing them in a way that the design cost for each one final component supplied in the market is significantly reduced while the component is much more efficiently utilized. In a way it is an attempt to achieve with smaller sections of the building what cannot be achieved with the whole building for reasons previously described.

The utilization of systems building has as an important side effect the dramatic reduction of the cost of the design process. In this case this reduction is achieved by automating the design process to a large extent, since the utilization of a previously designed building system relieves the project architect of the need to make these decisions which are already much more efficiently designed and allow him to be concerned with other aspects of the design.

Innovation with regard to computer applications in the building industry has been occurring consistently over the past twenty years, but initially their introduction had aped the traditional piece-meal introduction of the innovations. However the major potential for economies resulting from the development of computer systems along side building systems is now being increasingly realized.

The combined utilization of building systems techniques and the digital computer can automate and facilitate the application of systems building in architecture to such an extent that architectural services can become potentially available to a much larger portion of the built environment which now cannot afford them.

More specifically one can name three fundamentally different approaches when attempting to introduce computers in the building process:

- The most obvious and modest approach is to accept the general framework of a conventional design and construction process but to develop various discrete application programs and systems which replace certain manual design procedures within that process.
- A more ambitious approach is to replace the traditional "paper format" design data bases by a comprehensive computer based building description system. In other words drawing boards are replaced with computer graphic terminals, and electronic processing of data substitutes for paper and pencil techniques. By integrating a wide variety of application programs with a building description system , an integrated computer aided design system can be developed. Several such

systems have been developed to date, with a considerable degree of sophistication few of which have been serving as production systems with considerable degree of success. Of these systems some are designed so as to be more suitable to deal with a rationalized traditional environment conceiving the building as comprised of standard details, while others are more suitable to deal with component based building systems, conceiving the building as an assembly of pre-determined physical elements.

- A still more ambitious approach is to substantially reorganize the whole design and construction process in order to take fullest advantage of the potentials of computer-aided design. This is sometimes referred to as computer computer-aided building.

This thesis investigates into the potential inherent in such an approach. Both systems building and computer aided building attempt to intervene in the building process and propose changes in the traditional sequence in order to increase its efficiency. Systems building provides the conceptual framework in order to attempt to rationalize the building process so as for it to be able to cope with the increasing demands imposed by our industrialized society. Computer Aided Building attempts to utilize the current developments of computer technology in the building process, proposing changes in the process that will lead to best utilization of the advantages of the information revolution.

The particular consideration in this document is the promotion of an open system approach allowing the selection of a wide variety of subsystems from the open market for the design of each project and not restricting the architect in the utilization of closed systems.

With the powerful and affordable tools provided by the new generation of computers these methods can penetrate even the smallest architectural firm and thus enable a much larger portion of the built environment to benefit from the valuable contribution of architectural services, even if this implies for these occasions, the modification of the traditional ans more resource consuming design process, in order to adapt to the new needs and tools.

The proposed scenario allows the utilization of previously designed building systems in the majority of the various building projects regardless of their size or the size of the firm that will design them.

Within this context, the parallel introduction of the notions of the Computer Based Building System and the Open Computer System for Architectural Practice leads to the provision of an appropriate computing environment for the application of Systems Building in Architecture.

A Computer Based Building System is conceived as the product of the parallel development of a building system and a computer based information system. It therefore differs substantially from any existing integrated CAAD system suitable for component building, since in these cases the computer systems were developed after the building system had been completed. This meant that these existing CAAD systems only facilitate the use of already existing building systems and therefore could be more accurately described as "computer aids for the usc of building systems". On the contrary a "computer based building system" has the following advantages:

- There is a constant interaction during the development stage between the teams working for the development of the various parts of the system leading to several potential modifications of the physical building system in order for the potential of the computer to be more efficiently utilized. This extra set of considerations for the design of the system is likely to lead to a more efficient and often much more user friendly and easy to operate, final product.

- The development of a "computer based" Building System as opposed to a conventional "paper based" building system not only implies that the building system will be more easily used but also ensures that the system to be produced will be more efficiently designed, not only because of the utilization of computer facilities in the process but, above all, because the environment will impose early in the design process the proper discipline and will ensure that the rules of the Building system will be explicitly stated and their implications will remain under control. Therefore great savings in resources are achieved and the final project is both more efficiently designed and much more easily used by the project architect.

However, in order for a C.B.B.S. to be utilized certain preconditions must exist, among which the most critical is the existence of a certain computing environment within the architectural firms which will enable its use without difficulty.

Thus, this document introduces the notion of an Open Computer System for Architectural Practice, which is an appropriate modular and integrated computing environment in an architectural firm. It is a realistic scenario for the efficient incorporation of the potential of the digital computer within an architectural firm for the acquisition and manipulation of information, based on existing hardware and software facilities as well as the probable resources available in an average firm.

This thesis is divided in three major sections. The first section contains background information and is of no use to readers with previous experience in Systems Building and Computer Aided Design. The second Introduces the notions of a C.B.B.S. and a O.C.S.A.P., describes their utility and interrelation and provides a scenario for the use of a C.B.B.S. within a O.C.S.A.P. This section provides the theoretical foundation for this document and is practically the core of this thesis. The third section is a description of several computer programs written within the Computer Resource Laboratory of the School of Architecture and Planning at MIT over the last 18 months, which clarify some of the principles and techniques presented in this document.

SECTION 1. BACKGROUND :

Information on Systems Building and Computer aids in Building

SECTION 1. BACKGROUND : Information on Systems Building and Computer Aids in Building

CONTENTS: 22

SECTION 1. BACKGROUND : Information on Systems Building and Computer Aids in Building

1. INTRODUCTION

This first section aims in the provision of the necessary information which will allow the user that does not have previous experience in these fields to briefly familiarize himself with the most crucial issues in order to follow the rest of this document. The expert reader is encouraged to move directly to section 2 where the main body of this thesis commences.

However, it must be stated that this introductory section in no way aspires to fully cover the vast area that these titles suggest. The information provided has been compressed as much as possible and has been provided in a way that serves the purpose of this document. There is extensive bibliography associated to these heavily researched issues and the reader is advised to look there for more information on these areas.

The first part of this section addresses the issue of the Systems Concept in Building and provides brief information in Systems Theory, Systems Building, as well as Building Systems Development. The second part of this section, provides information related to Computer Aids in the Building Process. It first addresses earlier developments in the area of Computer Aided Architectural Design as well as integrated computer systems appropriate for Systems Building and then describes more recent developments in the area of Artificial Intelligence and Knowledge Engineering.

2. THE SYSTEMS CONCEPT IN BUILDING

2.1 Systems Theory

2.1.1 Introduction to "system"

The systems concept was first developed by scientists. Traditional analytical techniques relied on the isolation of the smallest possible component of the object under study and therefore failed to provide an accurate description of the whole when strong and complex interrelations between components existed. Subjects which had to be examined as organized wholes were given the general name "systems". (9)

A variety of definitions exist, postulated by specialists in various different fields:

- "A kit of parts with a set of rules to yield some desired behavior".(3)
- "A set of parts to accomplish a set of goals".(5)
- "An array of components designed to accomplish a particular objective".(6)
- "The assemblage of sybsystems or components united by some regular interaction or interdependence aims at the understanding of system as a functional whole".(2)

One can observe inherent similarities among these definitions of the word "system":

- System consists of parts (components - subsystems).
- Parts interact with each other are assembled into system and affect the performance of the system as a whole.
- System as a functional whole is to achieve a particular goal.

There are five considerations to be regarded when studying a system.(4)

- Total objectives of the system have to be well defined and carefully stated and are unrelated to the systems performance.
- Environment of the system is what lies outside,-the givens- of the system. The system is not isolated but it is surrounded by other systems, which, although in some extent influence and determine

how the system performs,are not affected by the system and therefore form part of the systems environment.

- Resources of the system is what lies "inside" the system. The system can change and use resources to its own advantage and for its own functions and therefore should use them as efficiently as possible.

- The components of the system or subsystems, correspond to the rational breakdown of tasks the system must perform. Within a defined system it is important to identify the tasks which each subsystem has to perform as well as the critical subsystems which have the greatest effect on total system performance on which effect one can base the determination of the subsystem hierarchy.

- The management of the system refers to the control of operations and interactions within the system, so that it will work as expected. The control not only implies the examination of system operation to assure that plans are carried out in accordance with original goals but also the evaluation of feedback, to improve system performance.

2.1.2 The Systems Approach

The systems approach is an orderly procedure followed by an interdisciplinary team in order to analyze and remedy problems (within their defined context) which leads to optimized results. The systems approach achieves this by avoiding traditional methods of independence or ad hoc treatment of the elements involved and by conceptualizing a process which utilizes many scientific and management techniques such as project management, system analysis and operations research.

According to Herbert C. Auerbach(1) the systems approach can be outlined as pragmatic, organized, empirical and theoretical.

It is pragmatic, since it is action oriented and its products must respond to real world needs and all activities within the system are oriented to meet these needs.

Organized, since the "system approach" method is primarily reliant upon organized managerial inputs and coordination of all components and process in the system. Information and resources, inputs into the system, are generally large and are required to be implemented and controlled by interdisciplinary teams, comprising of specialists, skilled professionals, system scientists, management scientists and others while the interaction of this team must also be controlled.

It is Empirical since lessons are to be learned from evaluation and feedback of previously implemented systems.

It is theoretical since in order to arrive to solutions, theoretical models can be built and extended with respect to the problems.

The proper technological and managerial applications of the systems approach are strictly speaking only those situations requiring serious consideration of the effects of interaction (between components) Systems analysis and systems engineering (and also related disciplines of operations research and management science) include techniques developed specifically to identify, measure, describe and control various kinds of interaction.(9)

Applications of the systems approach range considerably from scheduling of toll-bridges, layout of production-mix for a company, to missions of the aerospace industry. This approach is apparently more useful on large scale projects,and several; proposals have been made to use this approach in urban renewal;, improving the nutrition problem of mankind, health systems, and many other problems.(7)

2.1.3 Steps in the Systems Approach

The systems approach is based on a set of ordering principles and procedural rules. It is characterized by certain procedures and by a certain sequence of steps or phases for attacking a problem. The procedural rules corresponding to the various steps of the systems approach are as follows:(10) Problem definition:

This step requires the conceptual isolation of the system to be studied. It consists of the determination and definition of the problems which are generally associated with needs. Systems boundaries are specified, thus the environment in which to work in as well as the resources of the systems are defined. Through this understanding of the system's context results a quantitative and qualitative statement of the disparities between the actual state of affairs and what is desired or ideal.

Goals:

Establishment of systems objectives in relation to those problems. Each subsystem within the system studied must be defined in terms of its components, activities, as well as interaction with other subsystems. This implies recognition of every aspect and act which involves the whole system as well as the fact that these sets of components will function to accomplish the systems objectives and contribute to the performance of the whole.

Analysis:

Generation of the greatest possible amount of information about the problem, goals, evaluation criteria and modeling and the quantitative and qualitative aspects of their components and relationships.

Synthesis:

Generation of alternative ways to achieve systems objectives. For each alternative measurement of system's performance and feasibility under system's context and constraints.

Selection:

The selected alternatives with their supporting evidence are presented to the decision-makers. They will evaluate these alternatives in detail and chose one which best accomplishes the system objectives. This should result in the determination of the solution which in the models most nearly meets the evaluation criteria.

Implementation:

The execution of the selected solution in the real world.

Feedback:

Testing and evaluation of the results, referring to the performance of the whole system and more specifically addressing the issue of how well the system has been developed and how well its actions can be monitored. This is for the purpose of validation and feedback to the system. As a consequence, decisions can be modified and better decisions can be made in each stages. This, in turn, will modify the solution and improve the subsequent results.

2.1.4 References

1. Auerbach, H.C., (1969) "The Systems Approach to Building : What is it?" in Lectures and Proceedings of a Series of Regional Conferences Held in Seven Cities in Canada, September 30 - November 6, 1969, Department of Industry, Trade and Commerce, Ottawa, USA p.12

2. Brandle, Kurt, (1976) "The Systems Approach to Building: A core of information," ASCA Learning Package, 1974-1976, p.7.

3. Brill, M., (1976) Lecture Notes on Systems in Brandle, Kurt, "The systems Approach to Building: A core of Information," ASCA Learning Package, 1974-1976 p. 7.

4. Churchman, C.W., (1968) "The Systems Approach," Dell Publishing Co., Inc ., New York.

5. Ibid., p. 29.

6. Johnson, R.A., W.T. Newell and R.C. Vergin, (1972) Operation Management,Boston: Houghton Mifflin Company. p.5.

7. Rittel, Horst, (1972) "On the planning Crisis: Systems Analysis of the First and Second Generations," in Bedriftsokonomen, No. 8, 1972, p. 390.

8. Sittipunt, Preechaya, (1984), "The Systems Approach to Building : A study of Systems Building Development", Masters Thesis, S.M.Arc.S., M.I.T.

9. Vilett, John, (1972) "The Relationship Between the Performance Concept and the Systems Concept," in Performance Concept in Buildings, Joint RILEM-ASTM-CIB Symposium Proceedings, Edited by Bruce Foster, Philadelphia, p. 210.

10. Ware, T.E., (1972) "Performance Specifications for Office Space Interiors,"in Performance Concept in Buildings, Joint RILEM-ASTM-CIB Symposium Proceedings, Edited by : Bruce Foster, Philadelphia, 1972, p. 361.

2.2 Systems Building

2.2.1 A Systems Approach to Building

The Systems Approach was first applied to building 20 years ago, first to housing and school programs and then to other building types, e.g. hotels, shopping malls, office buildings etc. This methodology was concerned with the total process of building, its context, management and resources. Total process here means every stage from the identification of the need or demand, to the completion of the building and its effective life use.

Thus, the whole building process, as well as its management and operation, was subjected to disciplined and scientific methods of planning, design, building procurement and construction, a fact which to a large extent is achieved by cooperation among the various professionals involved in the building process. This brings into play the main characteristics of Systems Building, i.e. coordination and the utilization of a scientific management system in order to define, analyze and realize the developments of buildings and building projects.

The system approach to building, is concerned with the integration of both process and product of building by the use of scientific management techniques involving organized research and development to result in a rational model for guided invention. It converts scientific knowledge into applied technology by means of goal oriented processes within clearly defined contexts and constraints of function, cost and time.(3)

One can indicate two main features in this approach: Goals are clearly stated in performance rather than prescriptive terms, and the interrelationships of subsystems within the overall system are explicitly emphasized.

Prerequisites for the application of the systems approach to building:(2)

There is a need to broaden the concepts of "building industry" and "professionals" so that they take in to account all activities involved in the entire building process and leave room for feedback in the process for improvement or modification.

The construction industry due to growing demands and new needs should support the development of a management system as a disciplined approach towards solving its problems. This will enable the industry to perform important areas essential to its productive growth:

- Handle large volume construction according to growing marker demands.
- Manage, evaluate and coordinate a broad interdisciplinary team required to deal with multi-faceted and multi-level problems inherent in the entire process.
- Measure and evaluate its own performance, resulting in modification and improvement of its methodology.

As stated in 2.1.3 the basic steps of the systems approach are : Problem definition, Establishment of goals, Analysis, Synthesis, Selection, Implementation and Feedback. The diagram (2.2.1) illustrates clearly the major aspects and stages involved in the application of the systems approach to building i.e. the systems building process.

At the outset "Systems Building" is conceived in the context of an assumed "market place" of supply and demand. This market comprises, on the one hand, the consumers or users of the future buildings, whose needs and requirements have to be identified and the resources available to meet them, on the other.(4)

To bring both parts into a satisfactory balance entails that each is expressed in measurable terms. Measurement is a vital consideration in the prediction and control of implementation. Identification of needs requires an understanding of people, their activities, the equipment they use, the spatial configuration necessary to accomodate these, and the environmental conditions to be satisfied. Most of these can be expressed in measurable terms of quantity and quality.(9)

This information can then be translated to describe the range of performances required on the built environment. The descriptions state how the solution must perform and not what it must be.

The resources available to meet specified needs include manpower skills, finances, materials, land, machinery, organization methodologies, administration and management.

With data assembled for both perceived needs and available resources, it is then possible to enter the stages of analysis and ultimately synthesis. This is the modeling phase, using the techniques such as cost analysis, structural calculations and perhaps even mathematical models to stimulate the performance of alternative solutions before evaluation and selection of one solution or solution range. Thus alternative possibilities are explored and generated.(5)

The alternative which is considered the most accurate assessment of the intended results will be selected under the guidance of the evaluation criteria, followed by the implementation stage. this is the manufacture of the parts of buildings and their assembly at the site. The result at this stage is not necessarily a building system, it can be an acceptable range of building systems to be tested or a process to be implemented, or even the design of a new institutional structure, e.g., financing, or planning etc.

Thereafter, buildings are occupied, used, and/or tested, and subsequently can be evaluated against the goals, objectives and predictions of the conceptual models.

2.2.2 Advantages of Systems Building

Herbert C. Auerbach,(1) stated vividly the advantage of "Systems Building" as the following:

> " The realization of the 'Management System' and the 'Systems Approach to Building' will make it possible for the industry to take full advantage of the diversified products and talent available on a competitive basis in a free market economy. It provides the vehicle through which the best manufacturing capabilities, professional services and new technology could be integrated to produce the optimum solution. "

The promises of systems building can be viewed differently and in any aspects. However, the most significant problems facing building process, design, construction are the main reasons for using systems building :

- Buildings as part of a system of alternative designs can be extremely flexible. Also compatible building production provides the performance ability to accommodate change in a pre- determined and technically coordinated manner.
- There are significant savings in construction time and material costs as result of the efficient assembly of subsystems and the use of new prebidding procedures.
- There are savings in design time through the use of performance specifications, instead of conventional prescriptive specification documents.
- Total costs are usually expected to be less than those of conventional construction, owing to the use of standardized products and

due to scheduled time savings in procurement and assembly.

Principally systems building should result in greater flexibility, higher production quality and efficiency, rapid scheduling, and economy in design and construction.(7)

Concluding, the application of the systems approach to building enables one to step back, examine existing situation, identify needs and available resources assign legitimate and appropriate goals and collect manipulate and analyze information that will be essential in order to address the problem in an efficient and rational way. The stage of synthesis that follows, based on the appropriate analyssis, identification of needs and definition of goals and utilizing scientific management techniques is likely to lead to proposals that encourage innovation and indicate objectively those solutions that most efficiently satisfy the identified needs, with the minimum waste of resources.

Thus, the application of the systems approach to building, results in higher degree of rationalization and more efficient utilization of applied technology and available resources.

2.2.3 Systems Building and Building Systems

The term system may be understood in two ways; as a verb or as a noun. As a verb it refers to a way of doing something. As a noun it describes a collection or set of objects and their dependent relationships. In other words a system may be a process (software) or a product (hardware); a "set of rules" as well as a "kit of parts". Coincidentally, the term "building" has exactly the same ambiguity. It may refer either to the construction process or to an artifact that results from that process.(8)

When applied to building a system may refer to an organization of activities, as in a prefabricating system or to an organization of physical elements, as in a structural system i.e.

"Systems Building" and "Building Systems".

The terms "Systems Building" and "Building Systems" need to be differentiated and are defined as follows:(6)

"Systems Building" is a process of project development dealing with its planning, design, procurement, production and erection in explicit steps and procedures. By means of Systems building, the building process is organized, analysed and realized as a whole.

A "Building System" is the organization of tasks, resources and parts which, when integrated in a pre-engineered manner, results in methods for the construction of buildings and the creation of environment.

While the systems approach to building (or systems building) relates to the way of achieving and of applying systems, a system for building (or building system) relates to a particular technical procedure of physical procurement and assembly.

"Systems Building" viewed as a building "process" should be able to respond to various context variables, as for instance, varied physical, economic, political, social and technical conditions that may exist in a Country or a specific situation. Conversely, "Building System" viewed as a building "product" or "hardware", is designed solely to work within a given context. The development of a specific prototypical "Building System" is the result of the application of "Systems Building" techniques which are principally "software". Systems Building if properly developed is supposedly applicable universally, although the specific hardware which results from its use may not be so.

The evolution of systems building as well as building systems has tended towards increasing rationalization of the building process and products, with the goal of more efficient organization of task, resources, and integrated building components, all combined in a pre-planned, pre-coordinated and pre-engineered manner.

2.2.4 References

1. Auerbach, H.C., (1969) "The Systems Approach to Building : What is it?" in Lectures and Proceedings of a Series of Regional Conferences Held in Seven Cities in Canada, September 30 - November 6, 1969, Department of Industry, Trade and Commerce, Ottawa, USA.

2. Ibid pp. 16-17.

3. Dluhosch, E. Unpublished Class Notes, M346/00911, on Building Systems," Department of Architecture, M.I.T., Cambridge, p.4.

4. Rojanavanich, Pisit, (1983) "Systems Approach for Housing Prototype Improvement Case Study : Thailand", Masters Thesis, S.M.Arch.S., M.I.T., June 1983, p.40.

5. Ibid. p.42.

6. Sittipunt, Preechaya, (1884) "The Systems Approach to Building : A study of Systems Building Development", Masters Thesis, S.M.Arc.S., M.I.T., pp. 26-27.

7. Ibid p.51.

8. Vilett, John, (1972) "The Relationship Between the Performance Concept and the Systems Concept," in Performance Concept in Buildings, Joint RILEM-ASTM-CIB Symposium Proceedings, Edited by Bruce Foster, Philadelphia, 1972, p. 208.

9. Building Systems Development Inc., (1969) "The Potential for Systems Building," prepared for the Post Office Department, Washington D.C., p.7.

2.3 Building Systems Development

2.3.1 Introduction to Building System

Generically speaking, the notion of building systems in its modern sense has its source in the radical change brought about by the industrial revolution and its subsequent effects on materials, building products, processes and design in construction. A distinction should be made between traditional construction systems which refer to the conventional process of assembling traditional building elements on the site by cutting, fitting, bending, etc., and industrialized building systems which refer to either fully compatible pre-engineered elements, or total building packages.

A building system simply means a kit of building parts with sets of rules for their assembly into total operating systems in order to yield some desirable level of performance.(1) The inter-relation and coordination among the various building parts of the Building System, allowing them to be assembled into a wide variety of building forms, indicates the difference from conventional buildings where the building parts have only the capability of being assembled into one configuration, i.e. the building for which they were designed. The value of a building system lies in the characteristics of its elements that are made compatible within the system.

Classification of Building systems is normally based on "type specific" rather than "material specific" consideration. This is because most building systems are capable of being realized by more than one material, and systems depending on a single material for their design are rare. Consequently the conventional way of classifying building systems is generally based on the type of structural support system, i.e. post and beam, panel system, box or volumetric (space enclosing) system. In addition any of the preceding systems can be classified as either "open" or "closed" system.(7)

Deciding among an approach for open or closed building system is an important issue for a systems building program. Indeed, during the last decades, two distinct but related approaches to the development of industrialized building have emerged, known as the "open" versus "closed" approach to industrialized construction. Along with the mechanization and rationalization of conventional / traditional construction practices and process, the evolution of open and closed systems was accompanied by progressive industrialization and prefabrication of components, elements and structural systems as well as complete, whole building systems.(4)

In a closed building system, its sub-systems, components and parts are compatible only with the other subsystems constituting that particular program and are not interchangeable or transferable to another system. The choice of a closed systems approach offers two possible alternatives in its system design:

i) System designers design the required building system in a completely prescriptive manner and have industry bid against system designs and specifications. This approach requires establishment of a large technical bureaucracy and limits the concept, technique and quality of the system designed, by the skill of the design team.

ii) A system team prepares a performance specification for requisite subsystems and requires the bidders to bid in closed teams for all subsystems, with a general contractor.

In an open Building System, its subsystems, components and parts are interchangeable with other systems. Implicitly, the desired interchangeability of subsystems and their use for numerous alternative combinations of plans and / or geometrical forms for projects varying in size and design is the reason which had led to the development of more or less open systems.

Subsystems of an open system are usually of different origin and can be arranged to form a number of compatible combinations, i.e., their use is not confined to a single system. he more "open" a building system is, the more its coordination principles allow for interchangeability of subsystems and components, and, by this characteristic, provide for increased planning flexibility and the possibility of variability during the life-time of the building.(2)

2.3.2 Sybsystems Compatibility and Systems Integration

The essence of a building system lies on its concept of building, comprised of a number of subsystems, which collectively form a whole system. Subsystems can be defined, physically integrated, dimensionally coordinated, installed series of parts which function as a unit without prescribed performance limits.

In any building system design, it is very important to define and document the general requirements and functions for each respective subsystem, in terms of the constraints imposed by the program and resources available and to define their generic properties in terms of performance standards.

In subsystem design and development, the integration of subsystems into a total system, which requires their mutual compatibility is of importance. The required compatibility and mandatory interfaces for each subsystem are principally stated in the performance specifications. The careful description of mandatory interfacing responsibilities between subsystems is the key to success in assuring high quality, cost and time performance, without resorting to the use of a closed system.

It should be clear that the essential qualities of a good system lie in its subsystem integration and compatibility. The criteria for achieving compatibility are interrelated performance characteristics, convention of physical interfacing, application of dimensional systems and modular coordination, respect for spatial and/or technical norms and standards, control of joints and interfaces, versatility of components' joinery and so called "by passing" systems.

In system design and procurement, a set of coordinating principles is therefore to be established as a basis to assure mutual compatibility compatibility between structure, wall/partitions, equipment/furniture, mechanical service, infrastructure and space. All elements which make up the subsystems must be capable of being integrated within the rule system of the modular grid and the dimensional criteria chosen, including dimensional range, the accommodation of all possible junctions and joints, tolerance allowances and handling ease.

A systems set of coordinating principles therefore, is comprised of the dimensional system, the basic module and modular increment, and planning grid. Once these principles are established, then it is possible to proceed with the design of subsystems and their components. Each element, e.g., structure, non-bearing elements, equipment/furniture, mechanical, electrical service can be conceptually designed separately. Within the discipline of modular system, it is made possible to design each as discrete, but still mutually compatible subsystems.(5)

In summary therefore it is mandatory for each subsystem to adhere to the rules of modular and dimensional coordination, including joint and tolerance allowances, and to conform with all engineering requirements, as well as applicable codes and regulations.

2.3.3 Management Coordination in System Design (6)

In order to develop an effective building system collaboration between building product manufacturers is essential. Diverse industrial groups and fragmented conventional practice are now called upon to cooperate much more closely with each other in order to solve the new problems that have emerged and require new products; problems that none of them has been equipped before to solve alone. In the systems approach, manufacturers, developers, contractors and sub-contractors are formed into a group with a management consultant as their coordinator. The success of building systems development relies upon industrial cooperation.

In practical terms, a coordinated group has been usually composed of two or more separate manufacturers from different product lines who have pooled some of their resources at managerial, sales and technical levels. Once established, one manufacturer may act as the group leader, or the group may choose to retain a consultant project manager to coordinate their activities.

Having formed the group and established the role of the manager, various advantages emerge:

i) The project manager is -say- four firms who speak to the owner with one voice. Queries can be made and answers analyzed from four viewpoints all at once.

ii) The advantages and disadvantages of any of the owner's requirements can be assessed, and changes can be requested responding to the needs of all the members of the group simultaneously.

iii) The net result of this approach is a decrease in the total bid cost, i.e., the group bid.

iv) To operate firms as a group is not useful only for reaching a satisfactory solution to system design, but stimulates a continuation of cooperation between the members of the group as an entity into the vital production and erection phases as well. If there is no pre-coordination, a system may fail due to continuous and uncoordinated modifications after the termination of the design stage.

Other ways to coordinate building systems development programs are possible which are not necessarily dependent upon an owner to prepare performance specification and invite bids. The initiative may also come from a design group with industrial participation invited on the "promise" of a market for their products. At any case however, needs for coordination are similar,and systematic formalization of the design and execution process in building require coordinated management and sound research for solving the complex problems of performance requirements.

2.3.4 Developing the Hardware of a Building System (3)

Building systems design is significantly different from individual building design, and thus traditional design procedures are not applicable. A failure to recognize this can lead to frustrating and costly development experiences. The difference between building systems development and traditional building design stems from two causes:

i) System solutions are not specific to any one building problem.

ii) Systems require sound research and development, major commitments during all phases of the process.

Systems building programs take one of two general forms:

- They can be primary or developmental systems building programs, developed from scratch. In this case new performance statements are written, large markets are organized to accommodate the new building products, developed by manufacturers a fact which results in the hardware procurement by manufacturers.

- They can alternatively be secondary systems building programs, exploiting the speed, economy, flexibility and quality of building systems programs already developed under the primary programs. These secondary programs need little research and development, and therefore they can thrive on smaller markets than developmental programs. This brings us to the notion of the "second generation building system" which is in fact a successful application of a developmental - first generation building system after appropriate alterations or modifications on a different context.

In the planning / design of a building system, each sub-system, its components, elements, pieces of equipment, service etc., must be conceived on the basis of overall system requirements, which are conceptual programmatic, practical and serve the goal established by the program for which satisfaction of user needs and requirements is a primary concern.

There are a number of basic considerations in building systems design, to mention a few:

- All system elements should favor as much as possible "open" combinations.

- A building system must be capable of expansion, both horizontally and vertically.

- Full flexibility of all service media i.e.ducts, pipes, wiring etc. is desirable, both vertically and horizontally, without unnecessary and undue modification of the basic structural elements.
- Provision should be made for suitable tolerance allowances arising from different production and / or assembly methods.

The editors of "industrialization Forum" have proposed guidelines for system hardware developments consisting of the following steps:

Step 1 : Form the System Development team.
Step 2 : Check-out the potential market.
Step 3 : Analyze the Building Types within the market.
Step 4 : Analyze and evaluate existing systems.
Step 5 : Commence System Design.
Step 6 : Make a formal check of the system design.
Step 7 : Start field Tests.
Step 8 : Make Major Management Decisions.
Step 9 : Start Production.

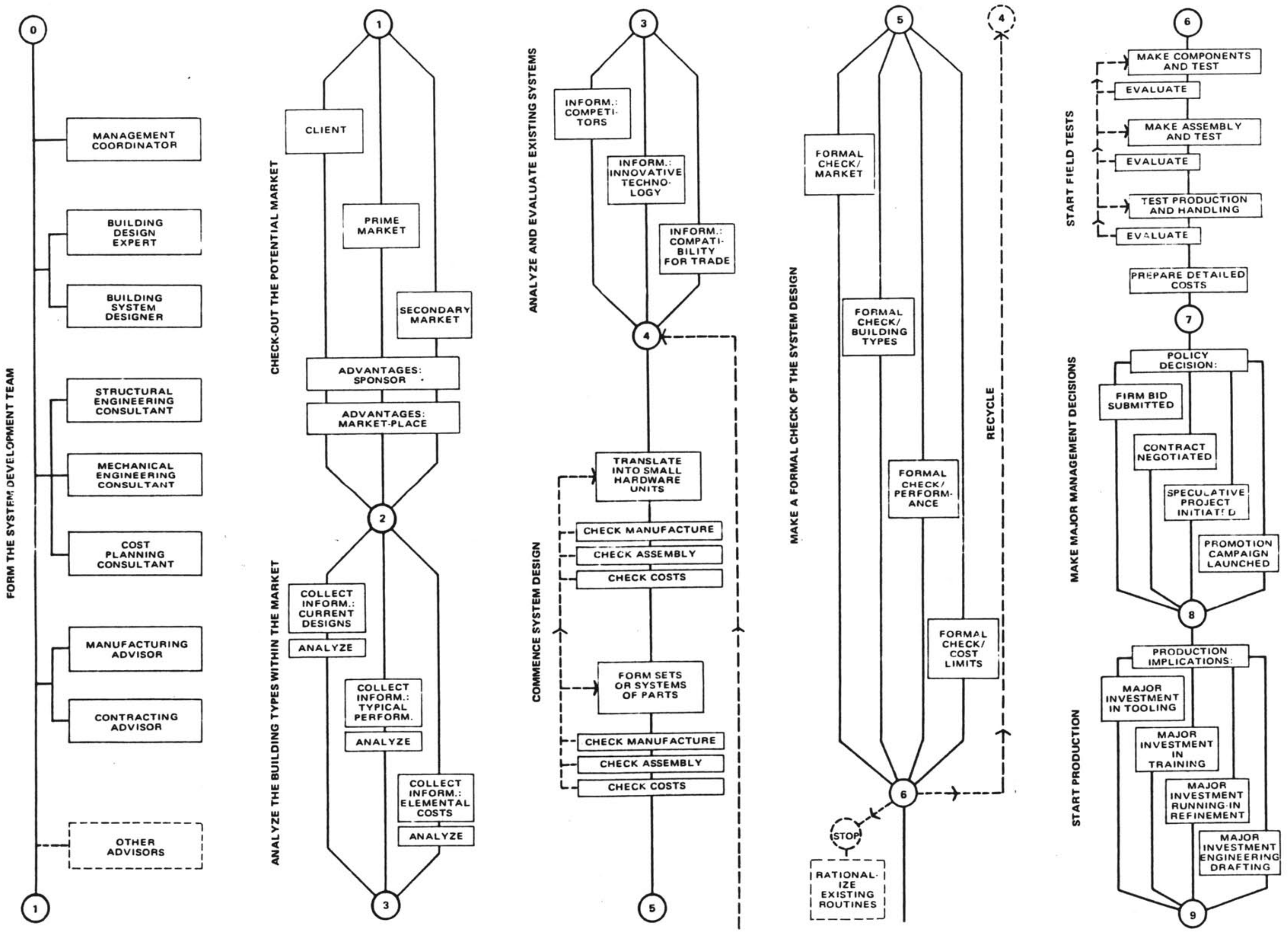
2.3.5 References

1. Bender, R., (1973) "A Crack in the Rear-view Mirror : A View of Industrialized Building", Van Nostrand Reinhold, p.63.

2. Brandle, Kurt, (1976) "The systems Approach to Building:A core of information.", ASCA Learning Package, 1974-1976, p.5.

3. I.F. (1970) "Developing the Hardware of a Building System", Guidelines prepared by the editor of I.F., in I.F., vol 1, No 3, 1970, pp. 23-28.

4. Dluhosch, E., Unpublished Class Notes, "Building Systems" M346/0091 p.1.



Developing the Hardware of a Building System. Source : the editors of IF vol.1, No 3, 1970.

5. Dluhosch, Erich, Unpublished Class Notes, Course 4.59 on "Building Systems II" Dept. of Architecture, M.I.T., Cambridge, Spring 1984, p. 13.

6. Hughes, Robert, (1969) "The Management Consultant in Systems Building," in I.F., Vol.1, No 1, 1969, pp.19-22.

7. Sittipunt, Preechaya, (1984) "The systems approach to building : a study of systems building development", Masters Thesis, S.M.Arch.S., M.I.T., p.128.

3. COMPUTER AIDS IN BUILDING

3.1 Computer Aided Architectural Design

3.1.1 Integrated Design Systems

A key consideration in identifying an integrated system is the location of the primary information store of the user organization. A design office, prior to its adoption of computing techniques, will have information stored in the heads of its staff, catalogues of product literature, files of drawings, and on library shelves. When computers are introduced into design offices they provide a further depository for information used by designers.

In executing any design function, a designer will call on information stored in any of these ways, and will abstract the subset of information relevant to the function, operate the function, access results and will use the resultant information to modify the preceding state of information. This pattern of information flow is familiar in fixed, function-centered information computing systems; these systems are set up to perform specific tasks, the designer then brings information to the system and assimilates results with all the other manual processes involved in designing buildings.

An integrated design system is based on a central database for the storage of all sorts of design data: it is intended to aid coordination and enforce consistency while enabling all members of the design team to work with the most recent information.

An integrated computer-aided design system consists of three components:

i) The design Database.

ii) Procedures for manipulating and interrogating the database.

iii) The hardware and software implementation.

The design database itself describes the project in terms of (a) "entities", (b) associated "attributes" and (c) "relationships" between the entities. The particular method of geometric description utilized in an integrated Computer Aided Design System is an important feature of the system, which determines to a large extent its performance and its suitability for particular applications. In principle, integrated systems are intended to serve as the primary information store to the user organization. The designer poses queries on a building description contained in a computer, identifies functions which are to be performed on the description, and the system is responsible for finding the data relevant to the query or function, and is responsible for assimilating results with pre- existing data. The principle aims in designing such a database are:

i) Comprehensiveness - it should describe all relevant aspects of the design.

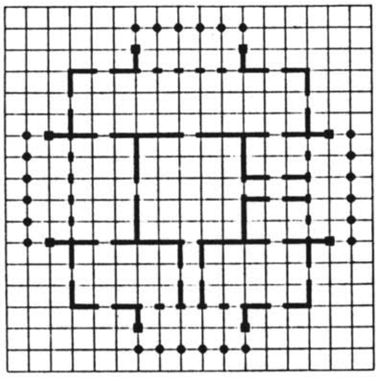
ii) Non-redundancy - information should be stored once only. This distinguishes the integrated system from a set of drawings where information is duplicated many times, and the possibilities for incompatibility are manifold.

iii) Consistency - the data should represent a feasible building, not a jumble of nonsence.

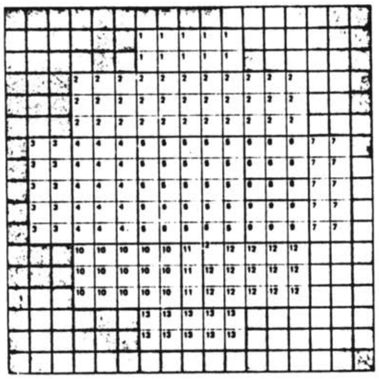
iv) Availability - the design database should be available to all disciplines all the time , so that everyone is working with the most up-to-date information.

Of these aims consistency is probably the hardest to pin down and achieve. It is semantic requirement which can only be defined in terms of the meaning of the data (unlike non - redundancy which is much more a logical property of the data structure).

The effects of such data-centered systems are evident in the reduced amount of data-preparation required from users, and in the reduced amount of data output which has to be vetted by users. Instead, these systems support a dialogue between designer and computer, relating the designer's interests to all the information stored in the machine, and enabling the designer to explore alternative design actions. Integrated, data-centered systems are dependent on a very close bond



(a) Scheme for a villa from J. Gwilt's Rudiments of Architecture (1826)

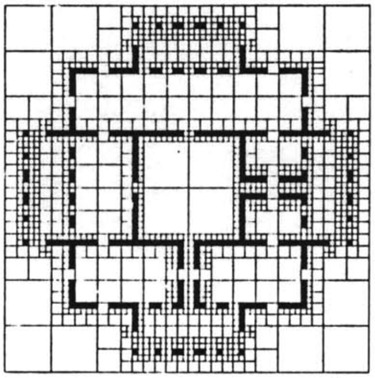


(b) Regular (square) grid representation

Advantages: Conceptual simplicity
Ease of implementation using array handling facilities of common programming languages
Spatial consistency is automatically maintained (since space is explicitly represented)
Many geometric computations, e.g. computing areas, testing for adjacency of spaces, are made simple

Disadvantages: Tends to consume excessive storage
Imposes a geometric discipline which may be inappropriate for many applications

Typical applications: Automated floor plan layout programs
Description of highly modular buildings

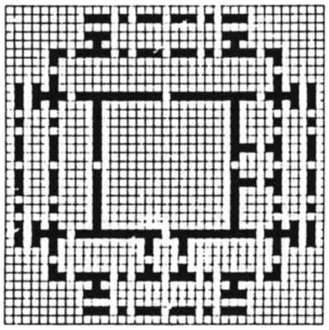


(c) Hierarchical array form of square grid representation

Advantages: Preserves many of the advantages of the simple square grid representation, while allowing relatively economical representation of fine detail

Disadvantages: Tends to be difficult to implement and manipulate, especially if the programming language employed does not incorporate the necessary facilities

Typical applications: No architectural applications reported
Mobile automata



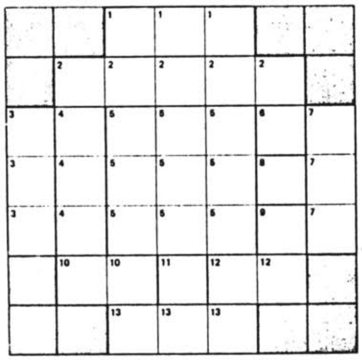
(d) Dimensionless representation (of solids and voids) in conjunction with dimensioning vectors

Advantages: Like the hierarchical array, preserves many of the advantages of the simple square grid representation, while allowing relatively economical representation of fine detail
Straightforward to implement and manipulate using array handling facilities of common programming languages

Disadvantages: Not well suited to description of non-rectilinear forms
Inefficient for description of very complex shapes

Typical applications: Comprehensive description of buildings of basically rectilinear geometry
Dimensional optimization programs

Comparison of different methods of geometric description. (part 1)
Source : William Mitchell, 1977, "Computer Aided Architectural Design." p 218

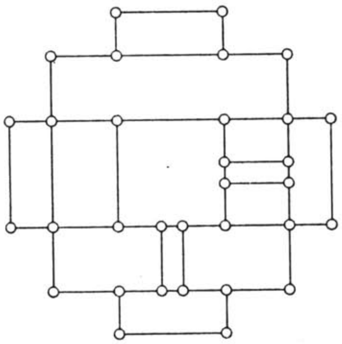


(e) Dimensionless representation (of spaces) in conjunction with dimensioning vectors

Advantages: Extremely efficient method for description of rectilinear forms
Preserves most of the advantages of the simple square grid representation
Separation of shape description from dimensional information facilitates efficient enumeration of generic plan forms

Disadvantages: Not well suited to description of non-rectilinear or complex shapes

Typical applications: Comprehensive description of buildings of basically rectilinear geometry
Automated floor plan layout programs
Dimensional optimization programs

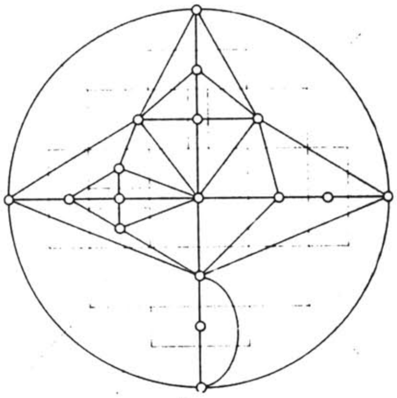


(f) Polygon (or polyhedron) representation

Advantages: Very general and flexible, imposes few restrictions on geometry
A large amount of research has been devoted to development of data structures and algorithms for handling this type of representation
Well suited to production of graphic output

Disadvantages: Sophisicated polygon or polyhedron representations tend to require the support of complex and extensive software
Implicit, rather than explicit representation of spaces causes serious difficulties in computing intersections, overlaps, adjacencies, etc.

Typical applications: Comprehensive building description
Graphics production applications
Spatial synthesis applications
Various engineering applications

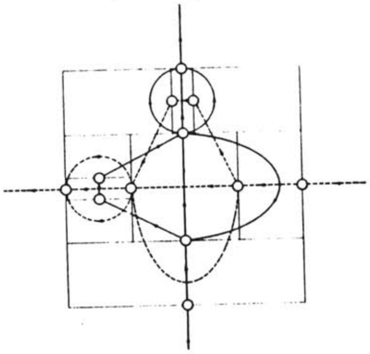


(g) Dual graph representation

Advantages: Directly and efficiently represents adjacencies between spaces
May be used as a basis for certain floor plan layout techniques
Can represent a floor plan concept before shape or dimension decisions have been made

Disadvantages: Essentially describes topology, and does not form a very convenient basis for developing a geometric description

Typical applications: Representation of very early layout concepts
Automated floor plan layout programs
Traffic or heat flow analysis



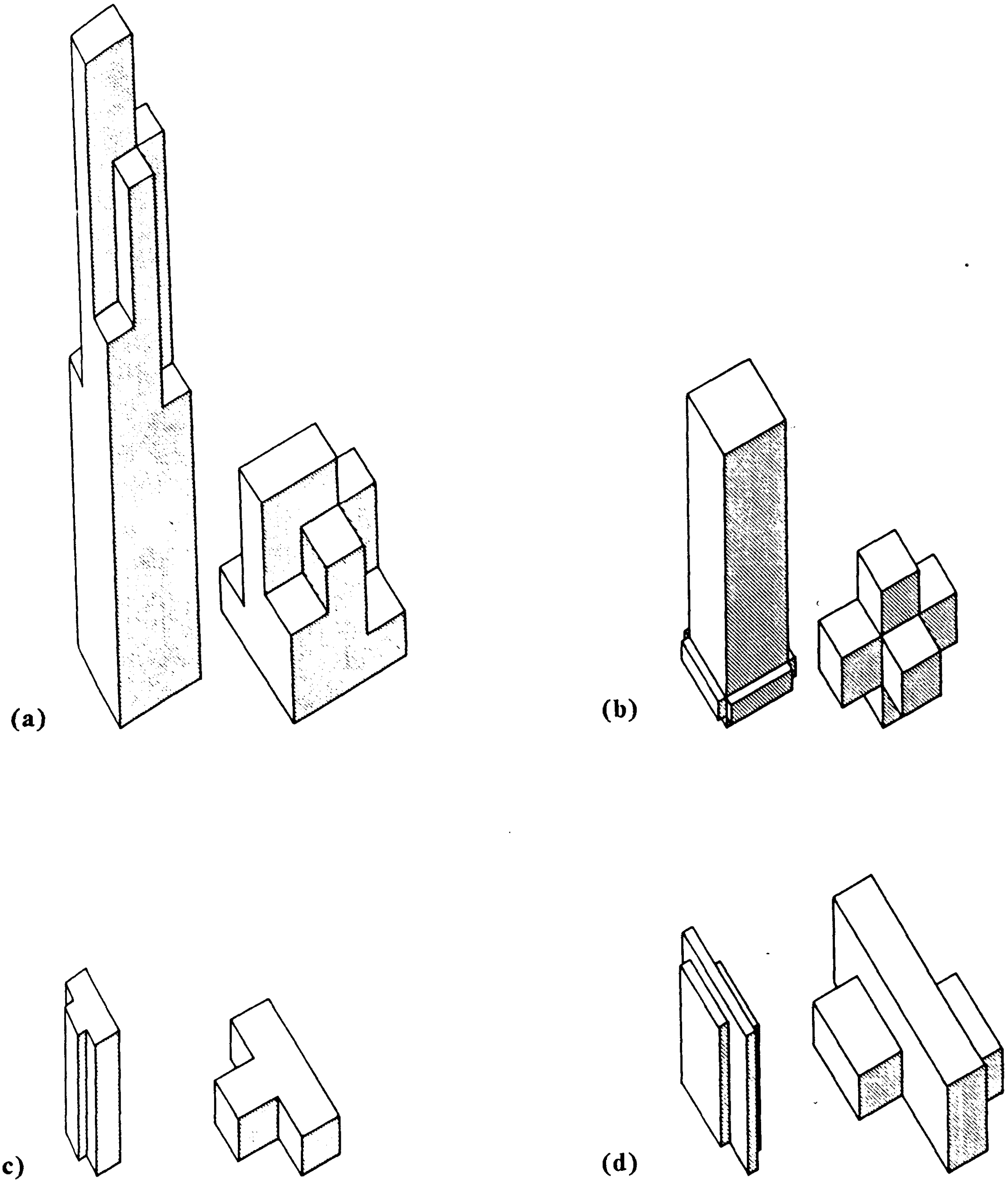
(h) Smith diagram (or Teague network) representation

Advantages: Very efficient for description of rectangular geometries
Allows exploitation of the "electrical network" analogy

Disadvantages: Restricted to rectangular geometries

Typical applications: Comprehensive description of buildings of basically rectangular geometry
Automated floor plan layout programs which exploit the "electrical network" analogy

Comparison of different methods of geometric description. (part 2)
Source : William Mitchell, 1977, "Computer Aided Architectural Design." p 219



Examples of high rise office buildings and their dimensionless representations. (a) Sears Tower, Chicago, (Skidmore, Owings and Merrill). (b) Place Victoria, Montreal (Luigi Moretti and Pier Luigi Nervi). (c) One Charles Center, Baltimore (Ludwig Mies Van Der Rohe). (d) Thyssen-Rohrenwerke office, Dusseldorf (Hentrich and Perschnigg). Source : William Mitchell (1977) Computer Aided Architectural Design. p 181 between the logical structure of a system and a user-organization's practices. The man-machine interface is crucial, and has to function successfully both in terms of shared knowledge available to the designer and the computer, and in terms of the vehicles which are used for conveying that knowledge, such as interactive graphics.

3.1.2 Criteria for Classification of C.A.A.D. Systems

Existing C.A.A.D. systems may be classified by the restrictions they place upon their project descriptions so as to simplify their implementation. The fact that a system incorporates a particular restriction is not a criticism of that system: it reflects the system designer's abstraction of a relevant feature of the context into which the system will have to fit. (1) The four restrictions which in various combinations, have so far been used by systems designers are:

1. Dimensionality : Most systems treat all spatial directions equally, and are thus fully three dimensional. However, some systems have regarded the vertical direction as subordinate to the two horizontal directions, and are thus termed " Two and half dimentional". Such a restriction, which may be appropriate for describing sites of single stories, permits many geometrical operations to be performed in 2D space and thus may lead to substantial simplifications.

2. Orthogonality : Most systems model the physical building as a set of cuboids, whose faces are each perpendicular to an axis. As compared to other systems which use some form of polyhedron as a modeling element, they benefit from both a simplification of all geometrical operations, and a significant reduction of storage space, since only six coordinates are required in order to specify the position and size of a paraxial cuboid. The use of paraxial cuboids as modeling element although permitting the system to describe non-orthogonal building geometries, imposes several limitations in this description since all interactions between elements must occur in paraxial planes and therefore foundations on sloping sites or junctions between roof pitches are forbidden.

3. Sparseness : Most systems are designed to describe arbitrarily complex buildings, and thus to process descriptions that contain large number of elements. In their design therefore much attention is paid to compressing the descriptions and eliminating redundant or superfluous information. On the other hand systems whose context requires only "sparse" descriptions with a limited number of elements are able to simplify the implementation of many operations by maintaining fully explicit, highly redundant descriptions of each element.

4. Discreteness : (16) A discreteness restriction constrains the attributes of an element to be unaffected by the addition, modification, or removal of the neighbouring element. In some cases, this restriction is re-inforced by factoring out from all instances of an element all information except position, orientation and element type. Systems intended for rationalised traditional context, on the other hand, cannot in general exploit this restriction because they have to represent elements whose attributes are largely determined by their relationship to their neighbours.

Based on the previously described criteria and according to the restrictions imposed to the user in order for the efficiency of the system to be increased, a set of particularly significant C.A.A.D. systems developed over the last 15 years may be classified as shown in the following table:

SYSTEM		DIMENSION	ORTHOGONAL	SPARSE	DISCRETE
OXSYS	(6,10,11)	3	YES	NO	YES
CEDAR	(3,4,5,13)	3	YES	NO	YES
BDS	(7,8,9)	3	NO	NO	YES
LRHD	(2,11)	3	YES	NO	NO
HD	(2)	2.5	YES	YES	NO
SL	(2)	2.5	NO	NO	NO
PIM	(15)	3	NO	YES	NO

It is interesting to note that each system embodies at least one of these restrictions and that most production systems embody two. Current research in C.A.A.D. is aimed at developing techniques in each of these areas which will permit systems to be based less onerous restrictions, and thereby to gain applicability in a wider range of contexts. The wide range of currently available C.A.A.D. systems in the market are presented through a set of tables (1) which provide a brief description of the essential features of each system.

3.1.3 References

1. ACADIA Conference Proceedings. (Spring 1985).

2. Bijl Aart, Stone David, Rosenthal David, (1979) "Integrated C.A.A.D. Systems" p(4-1 - 4-2)

3. Braid I.C. (1975) Six Systems for Shape Description and Representation - A Review" University of Cambridge CAD Group document 87, may 1975.

4. Chalmers J. (1972) "The Development of CEDAR" Proceedings of International Conference on Computers in Architecture, York, 1972.

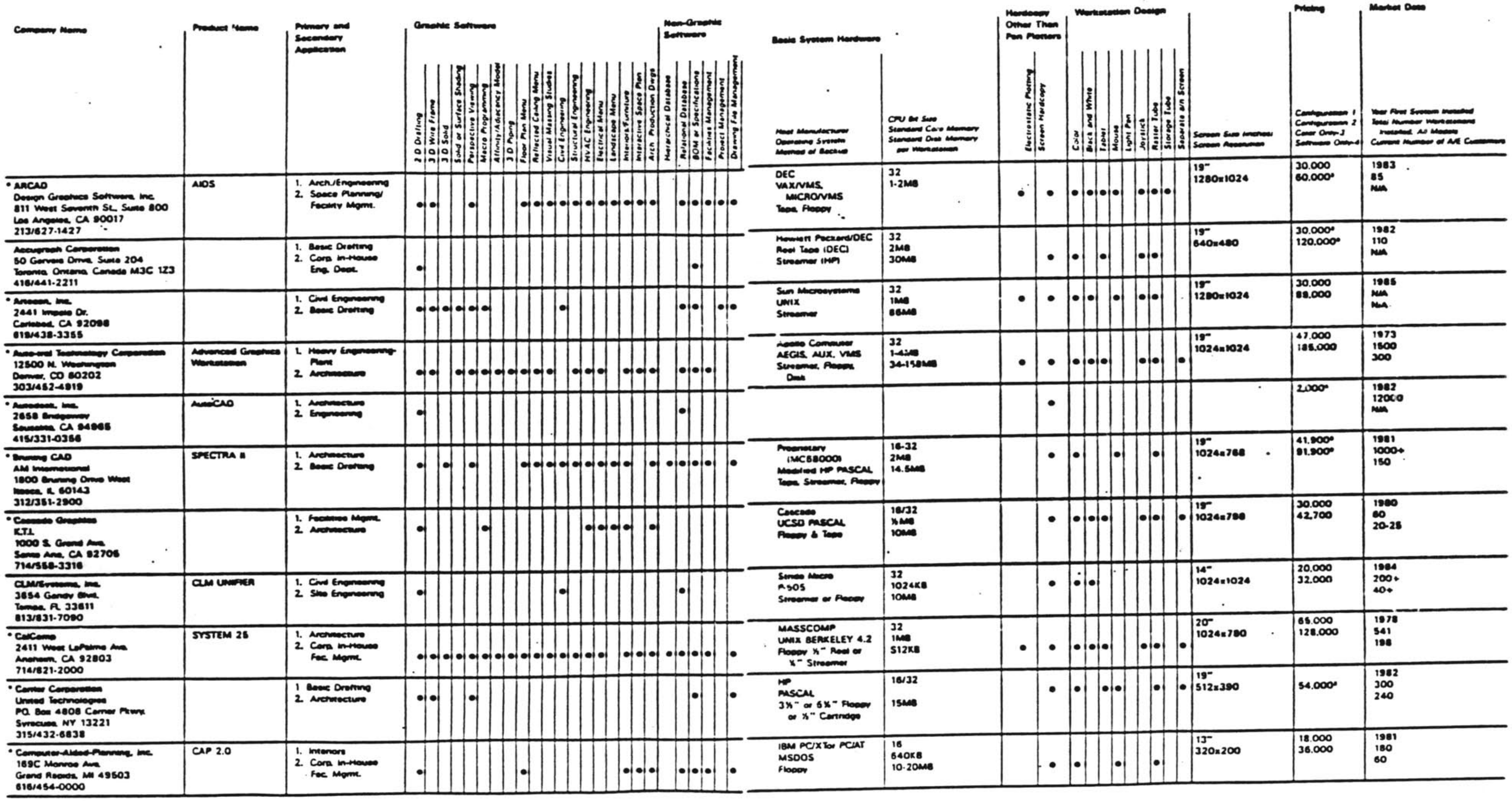
5. Chalmers J., Sampson P. & Webster g. (1974) "Data Structure Used in CEDAR" in Programming Techniques in CAD" Sabin. M. (ed) National Computer Centre.

6. Cross N. (1977) "The Automated Architect" Pion Limited. London.

7. D' Arcy Richard (1979) "The Oxford System and its application." Proceedings of the International Conference on Computers in Architecture.

8. Eastman C. et al (1974) "An Outline of the Building Description System" Inst. for Physical Planning, Carnegie-Mellon University.

9. Eastman C. (1975) "Preliminary Documentation for BDS-10" Inst. for



Company Name	Product Name	Primary and Secondary Application	Graphic Software																			Non-Graphic Software					
			2 D Drafting	3 D Wire Frame	3 D Solid	Solid or Surface Shading	Perspective Viewing	Macro Programming	Affinity/Adjacency Model	3 D Piping	Floor Plan Menu	Reflected Ceiling Menu	Visual Massing Studies	Civil Engineering	Structural Engineering	HVAC Engineering	Electrical Menu	Landscape Menu	Interiors/Furniture	Interactive Space Plan	Arch Production Dwgs	Hierarchical Database	Relational Database	BOM or Specifications	Facilities Management	Project Management	Drawing File Management
• ARCAD Design Graphics Software, Inc. 811 West Seventh St., Suite 800 Los Angeles, CA 90017 213/627-1427	AIOS	1. Arch./Engineering 2. Space Planning/ Facility Mgmt.	●	●			●				●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●
Accugraph Corporation 50 Gervais Drive, Suite 204 Toronto, Ontario, Canada M3C 1Z3 416/441-2211		1. Basic Drafting 2. Corp. In-House Eng. Dept.	●																					●			
• [illegible], Inc. 2441 Impala Dr. Carlsbad, CA 92098 619/438-3355		1. Civil Engineering 2. Basic Drafting	●	●	●	●	●	●						●									●	●		●	●
• Autotrol Technology Corporation 12500 N. Washington Denver, CO 80202 303/452-4919	Advanced Graphics Workstation	1. Heavy Engineering-Plant 2. Architecture	●	●		●	●	●	●	●	●	●	●		●	●	●		●	●	●		●	●	●		
• Autodesk, Inc. 2658 Bridgeway Sausalito, CA 94965 415/331-0356	AutoCAD	1. Architecture 2. Engineering	●																				●				
• Bruning CAD AM International 1800 Bruning Drive West Itasca, IL 60143 312/351-2900	SPECTRA II	1. Architecture 2. Basic Drafting	●		●		●				●	●	●	●	●	●	●	●	●		●	●	●	●	●		●
• Cascade Graphics K.T.I. 1000 S. Grand Ave. Santa Ana, CA 92705 714/558-3316		1. Facilities Mgmt. 2. Architecture	●					●								●	●	●	●		●						
CLM/Systems, Inc. 3654 Gandy Blvd. Tampa, FL 33611 813/831-7090	CLM UNIFIER	1. Civil Engineering 2. Site Engineering	●											●									●				
• CalComp 2411 West LaPalma Ave. Anaheim, CA 92803 714/821-2000	SYSTEM 25	1. Architecture 2. Corp. In-House Fac. Mgmt.	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●		●
• Carrier Corporation United Technologies P.O. Box 4808 Carrier Pkwy. Syracuse, NY 13221 315/432-6838		1 Basic Drafting 2. Architecture	●	●			●																	●			●
• Computer-Aided-Planning, Inc. 169C Monroe Ave. Grand Rapids, MI 49503 616/454-0000	CAP 2.0	1. Interiors 2. Corp. In-House Fac. Mgmt.	●								●								●	●	●		●	●	●		●

Company Name	Basic System Hardware		Hardcopy Other Than Pen Plotters		Workstation Design										Pricing	Market Data
	Host Manufacturer Operating System Method of Backup	CPU Bit Size Standard Core Memory Standard Disk Memory per Workstation	Electrostatic Plotting	Screen Hardcopy	Color	Black and White	Tablet	Mouse	Light Pen	Joystick	Raster Tube	Storage Tube	Separate a/n Screen	Screen Size (inches) Screen Resolution	Configuration 1 Configuration 2 Color Only-3 Software Only-4	Year First System Installed Total Number Workstations Installed, All Models Current Number of A/E Customers
• ARCAD	DEC VAX/VMS, MICRO/VMS Tape, Floppy	32 1-2MB	●	●	●	●	●	●		●	●	●		19" 1280x1024	30.000 60.000°	1983 85 N/A
Accugraph Corporation	Hewlett Packard/DEC Reel Tape (DEC) Streamer (HP)	32 2MB 30MB		●	●		●			●	●			19" 640x480	30.000° 120.000°	1982 110 N/A
• [illegible], Inc.	Sun Microsystems UNIX Streamer	32 1MB 86MB	●	●	●	●		●		●	●			19" 1280x1024	30.000 88.000	1985 N/A N/A
• Autotrol Technology Corporation	Apollo Computer AEGIS, AUX, VMS Streamer, Floppy, Disk	32 1-4MB 34-158MB	●	●	●	●	●			●	●		●	19" 1024x1024	47.000 185.000	1973 1500 300
• Autodesk, Inc.				●											2.000°	1982 12000 N/A
• Bruning CAD	Proprietary (MC68000) Modified HP PASCAL Tape, Streamer, Floppy	16-32 2MB 14.5MB		●	●			●			●			19" 1024x768	41.900° 81.900°	1981 1000+ 150
• Cascade Graphics	Cascade UCSD PASCAL Floppy & Tape	16/32 ½ MB 10MB		●	●	●	●			●	●		●	19" 1024x798	30.000 42.700	1980 60 20-25
CLM/Systems, Inc.	Senso Micro P-SOS Streamer or Floppy	32 1024KB 10MB		●	●	●								14" 1024x1024	20.000 32.000	1984 200+ 40+
• CalComp	MASSCOMP UNIX BERKELEY 4.2 Floppy ½" Reel or ¼" Streamer	32 1MB 512KB	●	●	●	●	●			●	●		●	20" 1024x780	65.000 128.000	1978 541 198
• Carrier Corporation	HP PASCAL 3½" or 5¼" Floppy or ¼" Cartridge	16/32 15MB		●	●		●	●			●		●	19" 512x390	54.000°	1982 300 240
• Computer-Aided-Planning, Inc.	IBM PC/XT or PC/AT MSDOS Floppy	16 640KB 10-20MB		●	●			●			●			13" 320x200	18.000 36.000	1981 180 60

Review of current "Computer Aided Design" systems. Chart 1. Source : ACADIA Conference Proceedings, Spring 1985.

Company Name	Product Name	Primary and Secondary Application	Graphic Software																			Non-Graphic Software						Basic System Hardware		Hardcopy Other Than Pen Plotters		Workstation Design										Pricing	Market Data
			2 D Drafting	3 D Wire Frame	3 D Solid	Solid or Surface Shading	Perspective Viewing	Macro Programming	Affinity/Adjacency Model	3 D Piping	Floor Plan Menu	Reflected Ceiling Menu	Visual Massing Studies	Civil Engineering	Structural Engineering	HVAC Engineering	Electrical Menu	Landscape Menu	Interiors/Furniture	Interactive Space Plan	Arch. Production Dwgs	Hierarchical Database	Relational Database	BOM or Specifications	Facilities Management	Project Management	Drawing File Management	Host Manufacturer Operating System Method of Backup	CPU Bit Size Standard Core Memory Standard Disk Memory per Workstation	Electrostatic Plotting	Screen Hardcopy	Color	Black and White	Tablet	Mouse	Light Pen	Joystick	Raster Tube	Storage Tube	Separate a/n Screen	Screen Size (inches) Screen Resolution	Configuration 1 Configuration 2 Color Only 3 Software Only-4	Year First System Installed Total Number Workstations Installed, All Models Current Number of A/E Customers
* Computervision, Corp. 100 Crosby Dr Bedford, MA 01730 617/275-1800	CDS 4000	1. Architecture 2. Engineering	●	●	●	●	●	●		●	●		●	●	●	●	●		●	●	●		●	●	●	●	●	Computervision Corp. CGOS/CVMOS Reel Tape	32 4MB Dynamic	●	●	●		●				●			19" 1024x1280	N/A	1970 N/A 120
	CDS 3000	1. Space Planning 2. Basic Drafting	●						●											●			●		●		●	CV (Lic. Sun Micro.) UNIX Cartridge	32 2MB 50 (to be increased)		●	●	●	●	●			●			19" 1152x900	N/A	1970 N/A 120
	PERSONAL DESIGNER	1. Architecture 2. Engineering	●	●		●		●														●						IBM PC-DOS Floppy	16 512KB 20MB			●		●				●			13" 640x400	17.890 52,680	N/A N/A N/A
Cymbal Cybernetics Corporation 169 Colonnade Road Ottawa, Ontario, Canada K2E 7J4 613/727-1880	multiDRAW™/AT	1. Basic 2D & 3D Drafting 2. Architecture	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	IBM PC AT PC-DOS 3.0 Floppy or Streamer	16 512KB 20MB		●	●	●	●	●			●			19" 1024x780	21.000 39.500	1983 42 15
Dhecha, Inc. 470 North Wiget Lane Walnut Creek, CA 94598 415/930-8762	1600PC Design Drafting System	1. Architecture 2. Facilities	●	●			●	●			●	●		●	●	●	●	●	●	●	●			●	●		●	IBM PC or Tektronics UCSD P-System or MS-DOS Floppy, Cartridge	16 512KB 10-40KB		●	●	●	●	●		●	●		●	19" 640x480	8.500 23.000	1981 95 200+
D.S.L. Inc. 411 West 7200 South, Suite 303 Midvale, Utah 84047 801/566-9238	DSL's SuperCAD	1. Architectural 2. Basic Drafting	●							●	●	●		●	●	●	●	●		●		●					●	IBM PC/XT & Compatibles MS PC DOS All	16 640KB 10MB		●		●	●				●			19" 1024x768	12.200 22.900	1984 33 24
* Datagraphic Systems 9101 General Drive Plymouth, MI 48170 313/451-7025	CAD MASTER 2000 Vers. 2.01	1. Corp. In-House Eng. Dept. 2. Architecture	●																					●			●	IBM MS DOS 2.01 or Higher Floppy	16 512KB 2MB		●	●		●	●			●			19" 1024x1024	14.780 N/A	1982 500 200
* Dignostics, Inc. Bohannan-Huston, Inc. 4125 Carlisle NE Albuquerque, NM 87120 505/884-3500	Digimap	1. Mapping/Civil Engineering 2. Facilities Management	●	●			●	●						●						●			●		●	●	●	DEC VAX Series VAX/VMS Reel Tape	32 4MB 456MB or greater	●	●	●		●				●		●	19" 1280x1024	135.000 185.000	1981 22 6
* ECOM Associates, Inc. 8634 W. Brown Deer Road Milwaukee, WI 53224 414/354-0243	A/E CADD 200	1. Structural Eng. 2. Corp. In-House Eng. Dept.	●					●						●	●			●	●		●			●	●		●	Hewlett-Packard Flexible Disk	32 2MB 16MB			●		●						●	19" 1000x1000	50.000 83.000	1984 2 1050
Engineering Systems Corporation 3636 S. Sherwood Forest Blvd. Suite 400 Baton Rouge, LA 70816 504/769-2226		1. Basic Drafting 2. Corp. In-House Arch. Dept.	●	●				●													●							Primarily DEC; others include MASSCOMP, CADMUS Floppy or Tape	16 or 32 256KB-1MB 10MB			●	●	●	●		●	●	●	●	12 to 17" 640x480 min. 4096x4096 max.	18.000 34.000	1978 400 390
* Foresight Resources Corp. 10950 Grandview Build. 34, Suite 170 Overland Park, KS 66210 913/345-2626		1. Basic Drafting 2. Corp. In-House Eng. Dept.	●																		●						●	IBM-PC, XT, AT MS-DOS 2.1 or later Floppy	16 512KB 384MB		●	●	●		●			●			12" 720x704	6.100 186.000	N/A N/A New
* GMW Computers, Inc. GMW Computers Limited (England) 1417 Fourth Avenue Seattle, WA 98101 206/467-0660	RUCAPS Building Modeling™ System	1. Architecture 2. Mechanical/Electrical/Piping	●	●	●	●	●		●	●									●				●		●	●	●	DEC or PRIME VMS Streamer or Reel Tape	32 1MB 60MB	●	●	●	●	●	●	●	●	●	●		19" 1280x1024	68.150-89.150 111.950-172.550	1977 118 N/A

Review of current "Computer Aided Design" systems. Chart 2. Source : ACADIA Conference Proceedings, Spring 1985.

Company Name	Product Name	Primary and Secondary Application	Graphic Software																		
			2 D Drafting	3 D Wire Frame	3 D Solid	Solid or Surface Shading	Perspective Viewing	Macro Programming	Affinity/Adjacency Model	3 D Piping	Floor Plan Menu	Reflected Ceiling Menu	Visual Massing Studies	Civil Engineering	Structural Engineering	HVAC Engineering	Electrical Menu	Landscape Menu	Interiors/Furniture	Interactive Space Plan	Arch. Production Dwgs.
* Graphic Horizons, Inc. (GHI) 60 State St. Boston, MA 02109 617/396-0075		1. Design Architecture 2. Facilities Mgmt.	●	●		●	●	●	●		●		●						●	●	●
* HASP, Incorporated 1411 W. Eisenhower Blvd. Loveland, CO 80537 303/669-6900	HASP Digital Survey Design System	1. Civil Engineering 2. Corp. In-House Eng. Dept.	●			●								●							
* HOK Computer Services Corp. Hellmuth, Obata & Kassabaum Arch. 100 North Broadway St. Louis, MO 63102 314/421-2000	HOKDRAW 2D/3D CAD Software In-house Developed	1. Architecture 2. Basic Drafting	●	●		●	●	●	●				●						●		●
* IBM Corporation Engineering CAD/CAM Market. 1133 Westchester Avenue White Plains, NY 10604 914/696-2177	CADAM with Design/Build/Manage	1. Corp. In-House Arch. Dept. 2. Corp. In-House FM Dept.	●	●				●		●	●								●		●
IGC Technology Corporation 305 Lennon Lane Walnut Creek, CA 94598 415/945-7300	PEGASYS	1. General Drafting 2. Architecture	●					●													●
* Information Displays, Inc. 11222 La Cienega Blvd., Suite 660 Inglewood, CA 90304 213/417-5386	IDRAW 3 (AEC Package)	1. Architecture 2. Basic Drafting		●		●	●			●											
Integrated Computer Tech., Inc. 1850 S. Amphlett Blvd. Suite 224 San Mateo, CA 94402 415/341-2946	IC200X	1. Facilities 2. Heavy Engineering	●					●			●	●		●	●	●	●	●			●
* Interactive Computer Systems, Inc. 9644 Brookline Ave. Baton Rouge, LA 70809 504/924-2081	1030 HS	1. Basic Drafting 2. Architecture	●	●			●	●		●	●			●	●	●	●	●	●	●	●
* Intergraph Corporation One Madison Industrial Park Huntsville, AL 35807 205/772-2000		1. All	●	●	●	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●
Lamont & Associates, Inc. 2775 West Hampden Avenue Englewood, CO 80110 303/761-8876	PROMISE	1. Civil Engineering 2. Land Development	●			●								●							
Maxam Technologies Inc. 8201 Corporate Drive Landover, MD 20785 301/731-7800	System 86	1. Architecture 2. Interiors	●	●			●	●	●	●	●	●				●	●	●	●	●	●

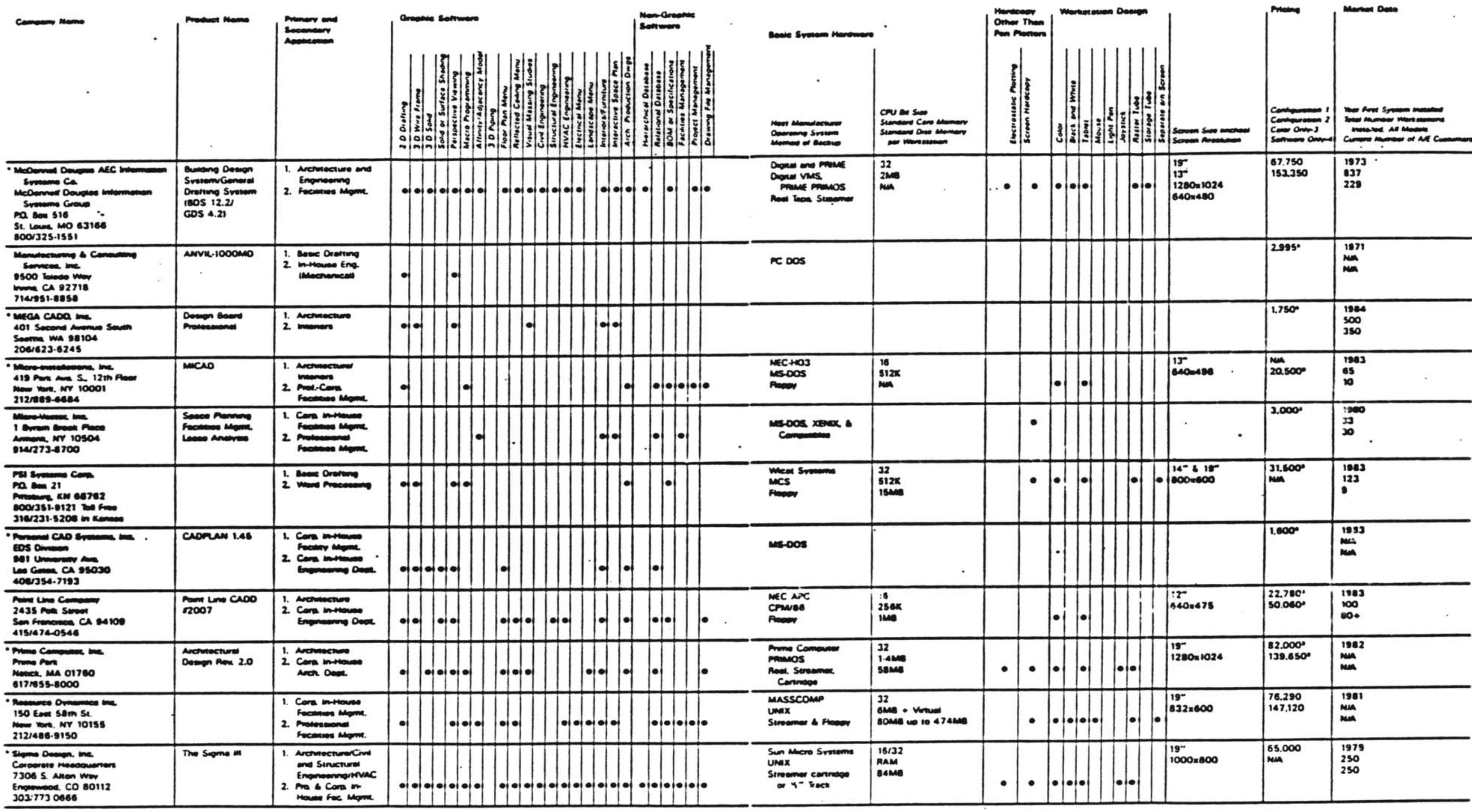
Company Name	Non-Graphic Software						Basic System Hardware		Hardcopy Other Than Pen Plotters	
	Hierarchical Database	Relational Database	BOM or Specifications	Facilities Management	Project Management	Drawing File Management	Host Manufacturer Operating System Method of Backup	CPU Bit Size Standard Core Memory Standard Disk Memory per Workstation	Electrostatic Plotting	Screen Hardcopy
* Graphic Horizons, Inc. (GHI)	●					●	PERQ Systems PERQ Op. Sy. (P.O.S.) Streamer	32 1MB RAM 35MB	●	●
* HASP, Incorporated						●	Hewlett Packard HPL-HASP Binary Choice of Floppy, Streamer, or Tape	16/32 524KB 15MB		
* HOK Computer Services Corp.		●		●		●	Digital VAX/VMS VMS Reel Tape, Streamer	32 1-8MB 60MB	●	●
* IBM Corporation	●					●	IBM Corp. VM/SP or MVS Tape	Varies Varies Varies	●	●
IGC Technology Corporation			●				Charles River Data Sys. UNIX Reel Tape, Streamer, Floppy	32 1MB 35MB		●
* Information Displays, Inc.							Apollo AEGIS Reel Tape, Floppy	32 1-4MB 86MB	●	●
Integrated Computer Tech., Inc.			●				IBM AT MSDOS Floppy, Streamer	16 640KB 20MB-32MB		●
* Interactive Computer Systems, Inc.			●	●			DEC DEC RT11 CALMA RSX 11 Reel Tape, Floppy	16-32 128KB 10MB, 40MB, 70MB	●	●
* Intergraph Corporation	●		●	●		●	DEC VMS Reel Tape	32 3-32MB N/A	●	●
Lamont & Associates, Inc.	●				●		PRIME PRIMOS Reel Tape	32 2MB None	●	
Maxam Technologies Inc.		●	●	●		●	Hewlett-Packard H-P Basic, UNIX Floppy	16/32 1MB 3MB	●	●

Company Name	Workstation Design										Pricing	Market Data
	Color	Black and White	Tablet	Mouse	Light Pen	Joystick	Raster Tube	Storage Tube	Separate a/n Screen	Screen Size (inches) Screen Resolution	Configuration 1 Configuration 2 Color Only-3 Software Only-4	Year First System Installed Total Number Workstations Installed, All Models Current Number of A/E Customers
* Graphic Horizons, Inc. (GHI)	●	●	●	●			●			19" 1280x1024	59.265 66.865	1982 68 39
* HASP, Incorporated	●	●	●				●			9" 400x300	41.350 44.500	1976 235 130
* HOK Computer Services Corp.	●	●	●	●						13"-19" 640x480 1280x1024	New New	1982 30 8
* IBM Corporation	●	●	●	●			●		●	19" 1024x1024	N/A N/A	N/A N/A N/A
IGC Technology Corporation	●		●			●	●			19" 1280x1024	62.950* 125.950*	1984 5 3
* Information Displays, Inc.	●	●	●							19" 1024x800	46.706 133.530	1977 150 8
Integrated Computer Tech., Inc.	●		●							19" 1024x1024	19.950* 42.000*	1983 100+ 40
* Interactive Computer Systems, Inc.	●	●	●			●	●		●	15 or 19" 800x600 1024x800 1280x1024	29.750 55.000	1980 300 N/A
* Intergraph Corporation	●	●	●	●			●		●	19" 1280x1024	100.000 N/A	1973 5877 N/A
Lamont & Associates, Inc.		●	●						●	14" 1024x768	N/A N/A	1984 8 25
Maxam Technologies Inc.	●			●			●		●	19" 512x390	39.950* 92.800*	New New New

Review of current "Computer Aided Design" systems. Chart 3. Source : ACADIA Conference Proceedings, Spring 1985.

Company Name	Product Name	Primary and Secondary Application	Graphic Software																			Non-Graphic Software						Basic System Hardware		Hardcopy Other Than Pen Plotters		Workstation Design										Pricing	Market Data
			2 D Drafting	3 D Wire Frame	3 D Solid	Solid or Surface Shading	Perspective Viewing	Macro Programming	Affinity/Adjacency Model	3 D Piping	Floor Plan Menu	Reflected Ceiling Menu	Visual Massing Studies	Civil Engineering	Structural Engineering	HVAC Engineering	Electrical Menu	Landscape Menu	Interiors/Furniture	Interactive Space Plan	Arch. Production Dwgs.	Hierarchical Database	Relational Database	BOM or Specifications	Facilities Management	Project Management	Drawing File Management	Host Manufacturer Operating System Method of Backup	CPU Bit Size Standard Core Memory Standard Disk Memory per Workstation	Electrostatic Plotting	Screen Hardcopy	Color	Black and White	Tablet	Mouse	Light Pen	Joystick	Raster Tube	Storage Tube	Separate a/n Screen	Screen Size (inches) Screen Resolution	Configuration 1 Configuration 2 Color Only-3 Software Only-4	Year First System Installed Total Number Workstation Installed, All Models Current Number of A/E
* SKOK Systems Incorporated 222 Third Street Cambridge, MA 02142 617/868-6003	Artech	1. Architecture 2. Professional Facilities Mgmt.	●	●		●	●	●			●	●	●	●	●	●	●	●	●	●	●		●	●	●			Hewlett-Packard HPBASIC Floppy, Streamer	32 2MB 1.2MB-14.8MB		●	●	●	●				●		●	19" 1024x780	29.900 91.900	1978 450 300
Starflo Engineering Starflo Corp. 4740 A Interstate Drive Cincinnati, OH 45246 513/874-4499	Code Petro Draft	1. Corp. In-House Engineering Dept. 2. Basic Drafting	●					●															●		●	●	●	DEC or IBM PC DEC VAX IBM MS-DOS Floppy	16 1MB 5MB			●			●						13" 640x400	N/A 30.000^{3}	1983 6 1
Summagraphics Corp. 777 State Street Extension Fairfield, CT 06430 203/384-1344	ICON	1. Basic Drafting 2. Civil Engineering	●	●			●	●						●							●			●			●	Data General DEC RDOS Floppy	16 512KB 15MB	●	●	●	●	●				●		●	19" 1280x1024	51.500 105.000	1978 300+ N/A
Summit CAD Corporation 5222 FM 1960 West, Suite 102 Houston, TX 77069 713/440-1468	Auto CAD/Plus Summit CAD Upgrade Package	1. Architecture 2. Engineering	●	●	●	●	●	●		●									●		●	●		●			●	IBM MS-DOS 2.7MB Floppy or tape	16/32 640KB 15 or 20MB	●	●	●			●						12" 640x400	12.000^{1} 32.995^{3}	1984 140 50
* Sys Comp Corporation 2042 Broadway Santa Monica, CA 90404 213/829-9707	Sys Comp Eng. Program Library Easinet CADD System	1. Civil Engineering 2. Structural Engineering	●	●	●	●	●	●														●	●					Data General Corp Data General AOS/VS Magnetic Tape	32 2MB		●	●	●	●				●			15" or 19" 640x480 1024x1024	N/A 65.000	1983 14 200+
Systemics, Inc. 2161 N.W. Military Hwy. Suite 214C San Antonio, TX 78213 512/344-4154		1. Architecture 2. Corp. In-House Architecture Dept.	●	●			●	●			●	●									●		●				●	Digital Equip. Corp. RSX-11M ¼" Tape Cartridge	16-32 1MB 35MB	●			●	●				●			13" 512x384	49.900 N/A	New New New
Technique Architectural Software 4438 Valencia Ave. North Vancouver, BC, CN V7N 4B1 604/980-5353	#9.38 for wide range of machines	1. Architecture 2. Space Planning	●	●		●	●				●		●					●	●	●	●	●	●		●			MS-DOS			●											2.000^{4}	1984 50 15
* T & W Systems, Inc. 7372 Prince Dr., Suite 106 Huntington Beach, CA 92647 714/847-9960	VERSACAD 4.0	1. Basic Drafting 2. Architecture	●	●			●												●					●			●	MS-DOS			●											2.000^{4}	1981 5000 1.800
* Torres Engineers Torres Engineers, Computer Applications Consultants 65 West Broad Street, Suite 702 Rochester, NY 14614 716/232-6995	CEDRA	1. Civil Engineering 2. Struc. Engineering	●	●			●	●						●	●								●				●	Prime PRIMOS Tape	32 1MB 34MB		●	●	●	●			●	●			19" 1280x960	70.000 72.000	1982 16 8
Tiltek Vision Systems 4710 University Way NE Suite 1512, Box C56789 Seattle, WA 98105 206/632-2125	EXECADD		●	●	●	●	●																					IBM PC, XT, AT														1.500^{4}	1984 N/A N/A
* VLSystems, Inc. 2691 Richter, Suite 118 Irvine, CA 92714 714/660-8855	VANGO (Ver. 4.2)	1. Civil Engineering 2. Surveying/Mapping	●				●							●								●			●		●	DEC VMS or RSTS/E Reel Tape	22 to 32 .5 to 4MB Varies	●	●	●	●	●	●					●	14" 768x293	35.000+ 43.000+	1980 450+ 35

Review of current "Computer Aided Design" systems. Chart 4. Source : ACADIA Conference Proceedings, Spring 1985.



Company Name	Product Name	Primary and Secondary Application	Graphic Software																			Non-Graphic Software					
			2 D Drafting	3 D Wire Frame	3 D Solid	Solid or Surface Shading	Perspective Viewing	Macro Programming	Affinity/Adjacency Model	3 D Piping	Floor Plan Menu	Reflected Ceiling Menu	Visual Massing Studies	Civil Engineering	Structural Engineering	HVAC Engineering	Electrical Menu	Landscape Menu	Interiors/Furniture	Interactive Space Plan	Arch Production Dwgs	Hierarchical Database	Relational Database	BOM or Specifications	Facilities Management	Project Management	Drawing File Management
* McDonnell Douglas AEC Information Systems Co. McDonnell Douglas Information Systems Group P.O. Box 516 St. Louis, MO 63166 800/325-1551	Building Design System/General Drafting System (BDS 12.2/ GDS 4.2)	1. Architecture and Engineering 2. Facilities Mgmt.	●	●	●	●	●	●	●		●	●	●	●	●	●	●		●	●	●	●		●		●	●
Manufacturing & Consulting Services, Inc. 9500 Toledo Way Irvine, CA 92718 714/951-8858	ANVIL-1000MD	1. Basic Drafting 2. In-House Eng. (Mechanical)	●				●																				
* MEGA CADD, Inc. 401 Second Avenue South Seattle, WA 98104 206/623-6245	Design Board Professional	1. Architecture 2. Interiors	●	●			●						●						●	●							
* Micro-Installations, Inc. 419 Park Ave. S., 12th Floor New York, NY 10001 212/889-6684	MICAD	1. Architectural/ Interiors 2. Prof.-Corp. Facilities Mgmt.	●					●													●		●	●	●	●	●
Micro-Vector, Inc. 1 Byram Brook Place Armonk, NY 10504 914/273-8700	Space Planning Facilities Mgmt. Lease Analysis	1. Corp. In-House Facilities Mgmt. 2. Professional Facilities Mgmt.							●										●	●			●		●		
PSI Systems Corp. P.O. Box 21 Pittsburg, KN 66762 800/351-9121 Toll Free 316/231-5208 in Kansas		1. Basic Drafting 2. Word Processing	●	●			●	●													●			●			
* Personal CAD Systems, Inc. EDS Division 981 University Ave. Los Gatos, CA 95030 408/354-7193	CADPLAN 1.46	1. Corp. In-House Facility Mgmt. 2. Corp. In-House Engineering Dept.	●	●	●	●	●				●								●		●		●				
Point Line Company 2435 Polk Street San Francisco, CA 94109 415/474-0546	Point Line CADD #2007	1. Architecture 2. Corp. In-House Engineering Dept.	●	●		●	●				●	●	●		●	●			●		●		●	●			●
* Prime Computer, Inc. Prime Park Natick, MA 01760 617/655-8000	Architectural Design Rev. 2.0	1. Architecture 2. Corp. In-House Arch. Dept.	●		●	●	●	●			●	●	●					●	●		●		●				●
* Resource Dynamics Inc. 150 East 58th St. New York, NY 10155 212/486-9150		1. Corp. In-House Facilities Mgmt. 2. Professional Facilities Mgmt.	●				●	●	●		●	●				●	●	●	●	●			●	●	●	●	●
* Sigma Design, Inc. Corporate Headquarters 7306 S. Alton Way Englewood, CO 80112 303:773 0666	The Sigma III	1. Architectural/Civil and Structural Engineering/HVAC 2. Pro. & Corp. In-House Fac. Mgmt.	●	●	●	●	●	●	●		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Company Name	Basic System Hardware		Hardcopy Other Than Pen Plotters		Workstation Design										Pricing	Market Data
	Host Manufacturer Operating System Method of Backup	CPU Bit Size Standard Core Memory Standard Disk Memory per Workstation	Electrostatic Plotting	Screen Hardcopy	Color	Black and White	Tablet	Mouse	Light Pen	Joystick	Raster Tube	Storage Tube	Separate a/n Screen	Screen Size (inches) Screen Resolution	Configuration 1 Configuration 2 Color Only-3 Software Only-4	Year First System Installed Total Number Workstations Installed, All Models Current Number of A/E Customers
* McDonnell Douglas AEC Information Systems Co.	Digital and PRIME Digital VMS, PRIME PRIMOS Reel Tape, Streamer	32 2MB N/A	●	●	●	●	●				●	●		19" 13" 1280x1024 640x480	67,750 153,350	1973 837 229
Manufacturing & Consulting Services, Inc.	PC DOS														2,995*	1971 N/A N/A
* MEGA CADD, Inc.															1,750*	1984 500 350
* Micro-Installations, Inc.	NEC-H03 MS-DOS Floppy	16 512K N/A			●		●							13" 640x496	N/A 20,500*	1983 65 10
Micro-Vector, Inc.	MS-DOS, XENIX, & Compatibles			●											3,000*	1980 33 30
PSI Systems Corp.	Wicat Systems MCS Floppy	32 512K 15MB		●	●		●				●		●	14" & 19" 800x600	31,500* N/A	1983 123 9
* Personal CAD Systems, Inc.	MS-DOS														1,600*	1983 N/A N/A
Point Line Company	NEC APC CPM/86 Floppy	16 256K 1MB			●		●							12" 640x475	22,780* 50,060*	1983 100 60+
* Prime Computer, Inc.	Prime Computer PRIMOS Reel, Streamer, Cartridge	32 1-4MB 58MB	●	●	●		●			●	●			19" 1280x1024	82,000* 139,650*	1982 N/A N/A
* Resource Dynamics Inc.	MASSCOMP UNIX Streamer & Floppy	32 6MB + Virtual 80MB up to 474MB		●	●	●	●	●			●		●	19" 832x600	76,290 147,120	1981 N/A N/A
* Sigma Design, Inc.	Sun Micro Systems UNIX Streamer cartridge or ½" Track	16/32 RAM 84MB	●	●	●	●	●			●	●			19" 1000x800	65,000 N/A	1979 250 250

Review of current "Computer Aided Design" systems. Chart 5. Source : ACADIA Conference Proceedings, Spring 1985.

Physical Planning, Carnegie-Mellon University.

10. Eastman C. (1975) "General Purpose Building Description Systems" Inst for Physical Planning, Carnegie-Mellon University Research Report 57 Sept 1975.

11. Hoskins E. "The OXSYS System" (1977) in Gero J. (Ed.) "Computer Applications in Architecture" Applied Science Pub.1977.

12. Mitchel William J. (1977) "Computer Aided Architectural Design" Van Nostrand Reinhold Company.

13. Patterson J.W.(1974) "An Integrated CAD System for an Architect's Department." Computer Aided Design, vol.6 no. 1. (Jan. 1974), pp.25-31.

14. Purcell Patrick A. (1975) "Computer-Aided Architecture in the United Kingdom," in N.Negroponte (ed.), Computer Aids to Design and Architecture. New York : Petrocelli/Charter.

15. Richens, P. (1974) "OXSYS: Computer-Aided Building for Oxford Method." Cambridge, England: Applied Research of Cambridge.

16. Rosenthal David S.H. (1977) "The PIM System - Integrated Display Facilities" EdC.A.A.D. Rept. 1977

17. Rosenthal David S.H. (1979) "Building description techniques for rationalised traditional Construction". Proceedings of the International Conference of Computers in Architecture.

3.2 **Computer Aids in Systems Building**

3.2.1 Introduction - Principles

The rapid and extensive development of computer aided design methods was undoubtly promoted by the rapid increase in accessibility of computers which happened in the mid-sixties. This accessibility was occasioned by developments in both hardware (cheaper, smaller computers, remote terminals, etc.) and software (english language programming, multi-access systems, etc.), so that by about 1967 access to a computer could be had quite easily at any conventional location. Thus in a space of only few years people no longer talked of computers in terms of the concept of "giant brains", but of the concept of the computer in the home.

At about the same time, rapid developments taking place in architecture considerably enhanced the relevance of the computer as an aid to architectural design. These developments were in the promotion and relatively widespread adoption of system building techniques - the use of standardized, prefabricated building components. The use of these techniques particularly suited computer-aided design methods because they meant that the architect was then dealing with a finite range of components whose attributes, i.e dimensions, costs, materials, strengths, etc., were known. (5)

As previously mentioned, when designing a Building System besides concentrating with the design of the components and parts of the system themselves, one has to a greater extent be involved with:

i) Designing the logic controlling the assembly of sets of components.
ii) Designing the interfaces between the components.
iii) Designing the rules governing the design of individual components.

These are all key characteristics of building systems from the standpoint of the development of computer applications. The individual building components as long as they conform to the logic stated by such rules are not intrincically important. The degree of simplicity with which such sets of rules can state the limits to the combinatorial possibilities of components represents the degree of coordination of the building method. The reduction of complex organizations to definable sets of rules is a familiar requirement to all those concerned with computer system design and complex systems analysis tasks. (9)

Thus, "component based" building systems made life a lot easier for the computer programmers, who could now establish dimentional grids, catalogues of components, etc., that a computer could easily handle. Instead of the virtually infinite freedom of design in traditional building, the architect working with system building was constrained within a system boundary which could also be managed by the machine. This strong link between computer aided design and system building remains effective, and most of the comprehensive CAD systems under development or in application are based on particular building systems.

It is important to distinguish between the integration of discrete application programs and the integration of a computer system with a building system. In the latter, we mean more than interfaces between discrete application programs, such that an updatable description of buildings is held in the computer and different programs can be run without inputting data anew: in an integrated system, applications can cover some or all of the aspects of the building process from the design stage to construction, and data in a component file can be called for evaluation or optimization routines. If the computer system is tied to a particular building system whose rules for assembly and choice of components are defined, there is considerable scope for automatic generation for the design and compilation of production information.

Each party in the building process - from those concerned with brief preparation of production information, through to those responsible for construction and cost control - is contributing to one entity : the building. Traditionally each party deals with its own application area while the architect attempts to retain overall coordination. One of the advantages of an integrated system is that information about the building is stored and coordinated centrally in the computer and parties concerned can access and up-date data, at any time and in the form most suited to their needs, thus reducing the chance of error and lack of communication.

The integration of discrete applications into packages, whilst enabling each party to carry out a better job within his sphere, tackles neither the problems of information flow between them nor the problem of communicating changes in design.

3.2.2 Integrated C.A.A.D. Systems for Systems Building

Numerous C.A.A.D. systems have been proposed and in many cases implemented. The earlier systems to be developed were severely limited in their capacities and capabilities by inadequacies in their building description facilities, and weeknesses in the programs which they incorporated. Many where nothing more than demonstration systems, capable only of handling very restricted "toy" problems. (15) But some of the more recent systems are sufficiently powerful to have achieved a high level of practical usefulness.

Most of these systems are dedicated to the development of a particular building type, or are suitable for a particular construction process. Some of them are further restricted to be used with a previously defined, already developed building system. Systems developed for such purpose tend to be much more powerful and easy to implement since they can impose discreteness restriction upon the users of the system. In general. Most systems impose one or often two of the previously described restrictions upon the user in order to increase their efficiency and there is a general trade-off among generality and efficiency of the system.

More recent systems like OXSYS (8,10) and CEDAR 3 (1,4,5,18) are particularly significant since they suggest an intelligent way to compromise between the immediate practical advantages of assuming a specific, narrow context of building type and construction system and the ultimate goal of achieving a high degree of generality and flexibility in automated architectural design systems. (15)

The brief description of the various significant C.A.A.D. systems previously developed which will follow, will in no way be comprehensive but will focus primarily in systems dedicated to computer aided systems building and attempt to provide the user with a continual sequence of developments and achievements in this area.

These systems have been particularly significant achievements at the time of their development and in several cases of later systems have been implemented with a great degree of success over a long period. They represent the closest examples of C.A.A.D. developments to the concept presented in this thesis and they have been tested in practice for several years, therefore serving as important precedents for any new attempt for innovation in the area.

However, these systems were all designed in order to facilitate the use of already developed building systems. Therefore the product of the integration of these computer systems with the pre-existing building systems - the computer aided building system - should not be confused with the "Computer Based Building System" that will be presented in the last part of this thesis. The latter implies the parallel development of the building system and the computer system.

3.2.3 Case Studies of C.A.A.D. Systems Related to Systems Building

The West Sussex System. (16,17,19)

The pioneering integrated system in Britain was developed by the west Sussex County design offices in Chichester. It was an interactive graphics system, employing refreshed cathode ray tube graphics terminals equipped with light pens and driven by an in- house IBM System/370 computer. West Sussex employed the SCOLA industrialized component building system, and utilized a serial tendering relationship with contractors whereby bids are made to produce a certain volume of construction of a certain type for a certain price before the buildings to be produced have actually been designed. This allows prices to be accurately known at the design stage. The computer system provided facilities for interactive graphic descriptions of buildings, and cost analyses of SCOLA building designs, plus environmental and structural evaluations, and automatically generated construction documentation. The building types handled included libraries, schools, and health care buildings. For a number of years the system was in everyday practical use, but by 1973 a change in policy had led to its discontinuation.(15)

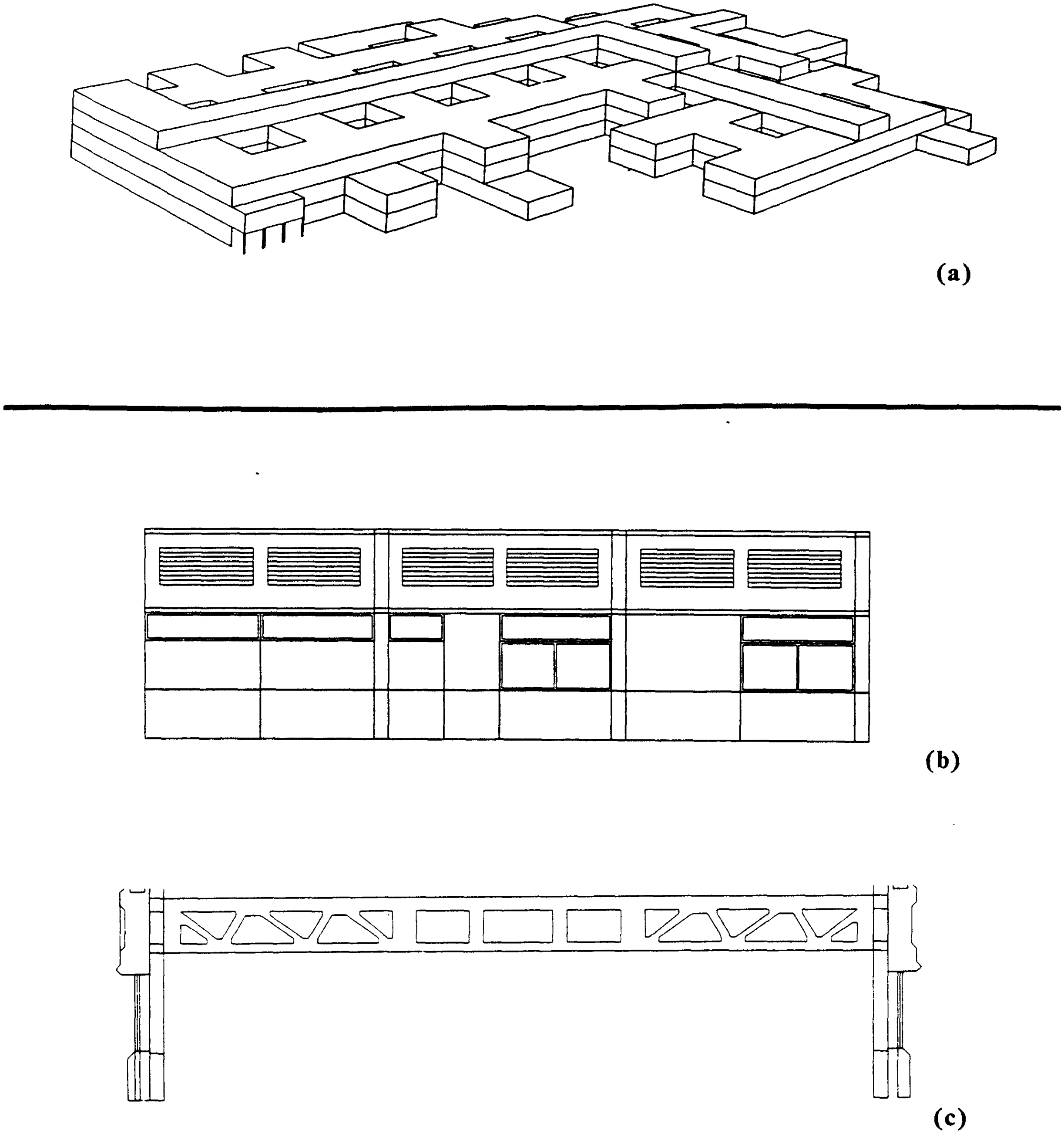
The Harness System. (11,13,14)

A more recent system, which also operates within the context of an unconventionally organized design design and construction process, is the Harness hospital design system, sponsored by the U.K. Department of Health and Social Security. A Harness hospital is assembled by arranging standardized, pre- designed hospital departments along a circulation spine.(15) Rules of assembly are defined for this kit-of-parts, so that the range of potential arrangements suitable for a given situation is well defined and relatively small. Several different building systems may be used for the construction of Harness hospitals. The computer system is not dependent upon interactive graphics, and due to the high level of standardization of Harness hospitals, the design process can be almost completely automated. The system performs tasks of structural, environmental and cost evaluation, automated generation of layouts, and production of documents.

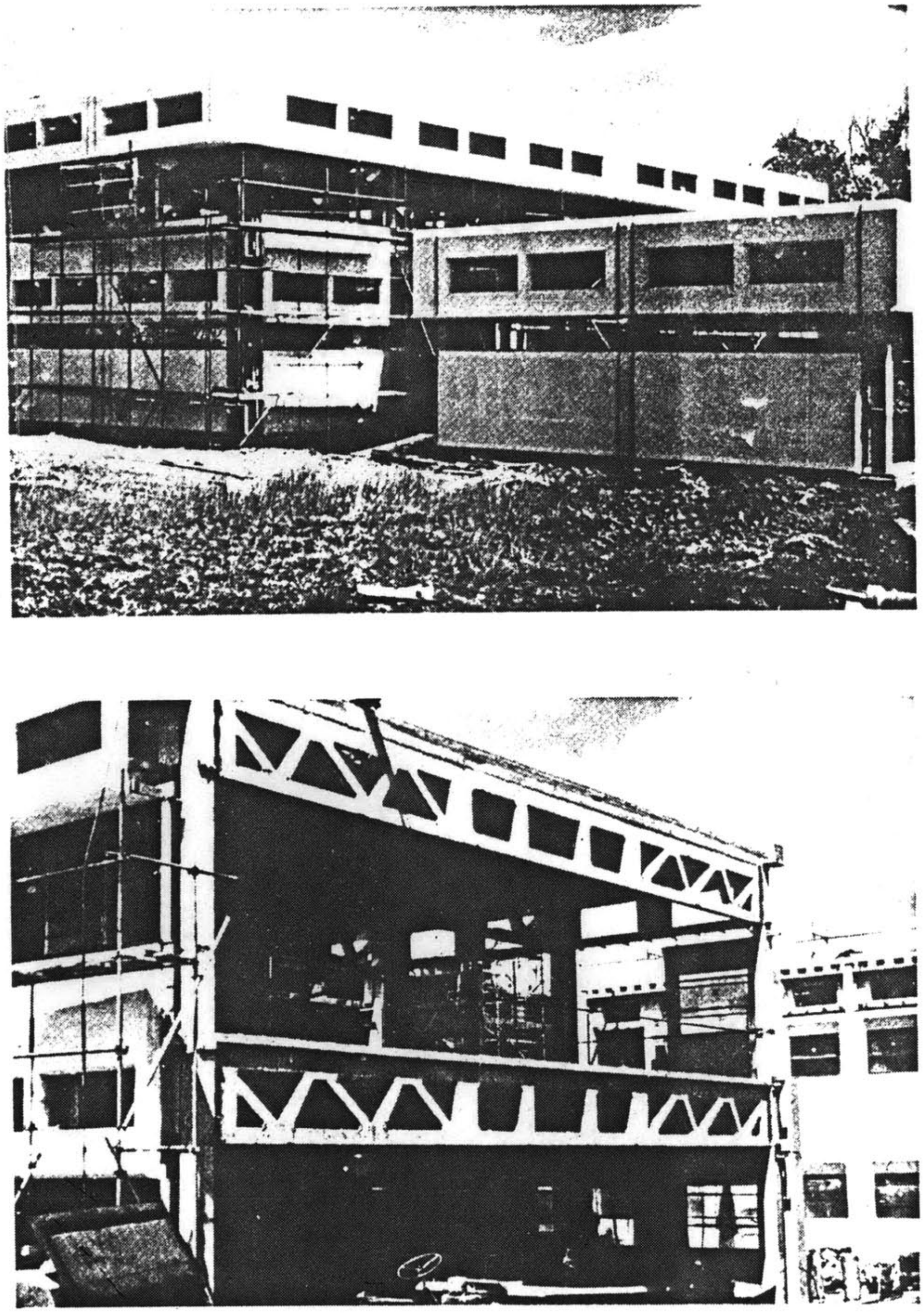
The CEDAR 2 System. (2,15) Cedar 2 was a large scale computer system sponsored by the Department of the Environment, utilizing interactive graphics and was oriented toward construction of post-office buildings using the SEAC (South East Architects Collaboration) component system. A pilot version was implemented in 1973. It provided capabilities for cost estimation, daylight, thermal, and acoustic analysis detailed design of framing and external walling, and production of documentation. Cedar 2 is basically a design evaluation and documentation system; the principal responsibility for design synthesis is allocated to human designers.

The CARBS System.

An other similar example of a C.A.A.D. system developed towards the same lines with CEDAR 2, is the CARBS (Computer Aided Rationalized Building System), developed by the Liverpool University

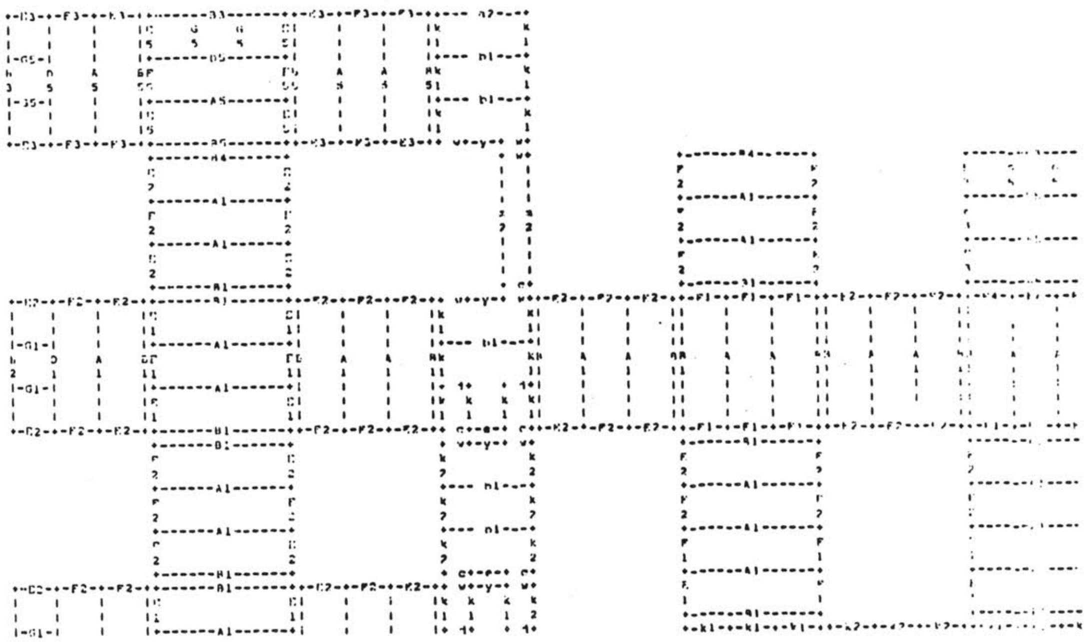


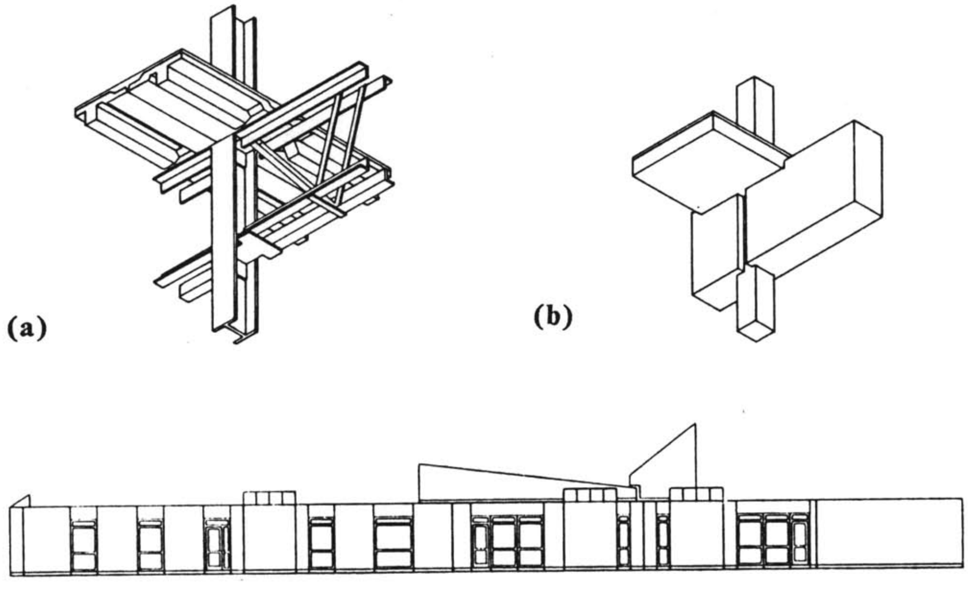
Harness Hospitals. (a) Computer - generated perspective of a typical Harness Hospital, showing arrangement of standard departments along a circulation spine. (b) structure. (c) Cladding. Source : William Mitchell (1977) Computer Aided Architectural Design. p 103



Harness Hospitals. (a) and (b) Prototype under Construction. Source : William Mitchell (1977) Computer Aided Architectural Design. p 104

(a)





Top : (a) An example of a structural plan aumatically generated by the Harness System Source :William Mitchell (1977) Computer aided Architectural Design p 243 **Bottom : Post-beam and pannel system (a) System of Construction. (b) Approximate representation of system geometry by orthogonal rectangular parallelepipeds. (c) Drawing, produced by te CARBS system, of an industrializedcomponent building.** Source : Mitchell (1977) p 105

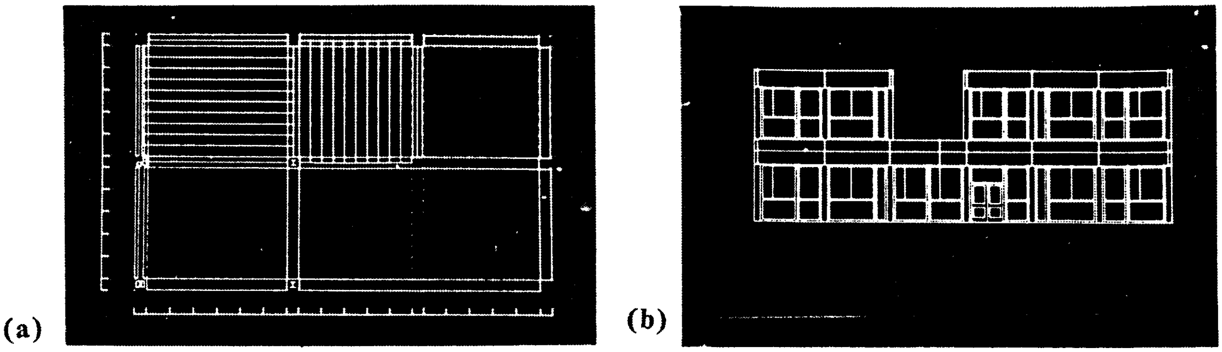
Computer-Aided Design Center in collaboration with Clwyd County, Architects Department (6)

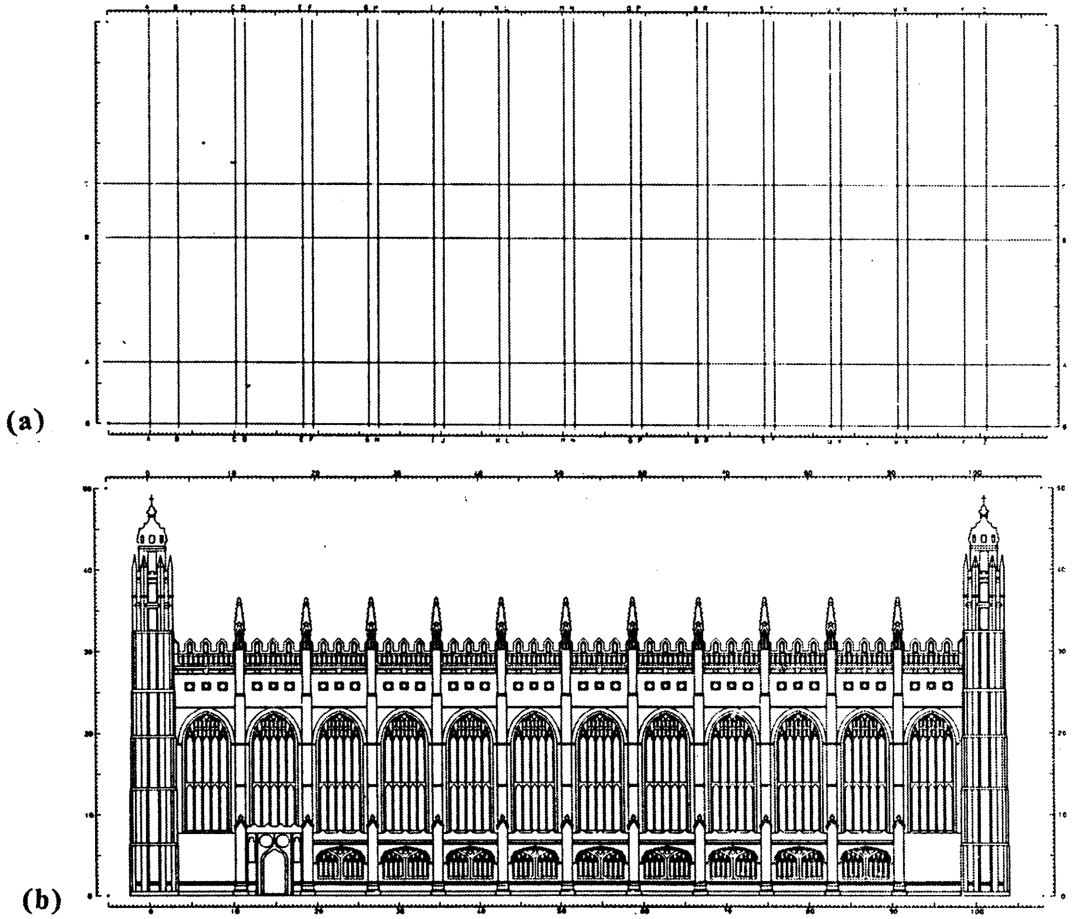
The OXSYS System. (8,9,10,15)

The OXSYS System was developed for design of hospitals in the Oxford method of Building. It incorporates capabilities for cost estimation from early sketch designs, performing structural and environmental analyses, semi-automatic design and detailing, and production of documentation. The system was put into production in 1974. The System has four obvious effects on a project : It increases Design resources, Design evaluation, Coordination, and Communication (7). OXSYS is designed so that it is not restricted to any one particular building system; component descriptions and assembly rules for different systems of the same general type may be loaded into the computer as data. The OXSYS approach appears to be an intelligent compromise between the immediate practical advantages of assuming a specific building type and construction system as exemplified by Harness and CEDAR 2, and the ultimate goal of achieving a high degree of generality and flexibility in automated architectural design systems.(15)

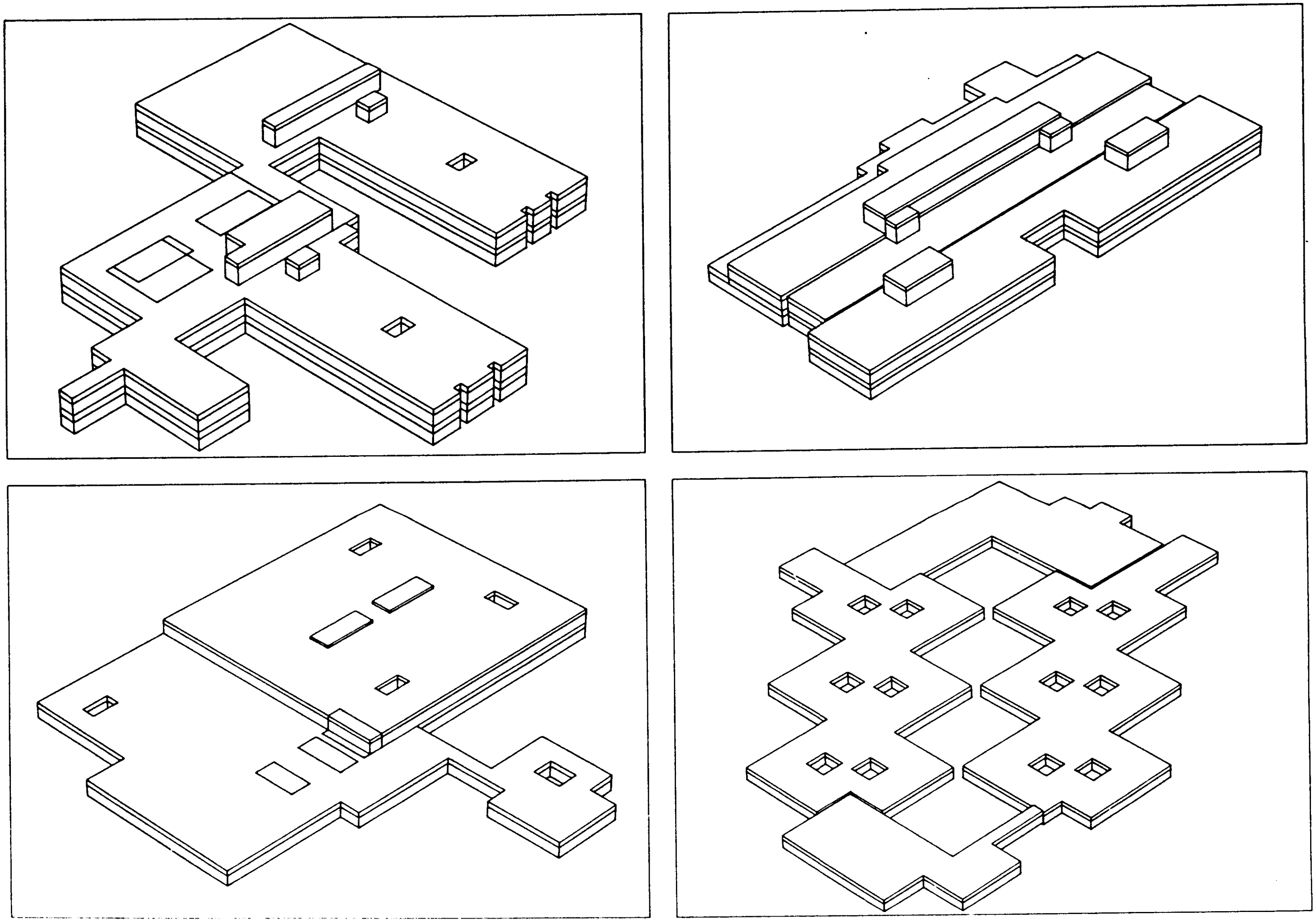
The CEDAR 3 System. (1,4)

Another fairly general system is CEDAR 3. Development work on this successor to CEDAR 2 was begun in 1975. It is intended for use in conjunction with the U.K. Government Property Services Agency's Method of Building (MOB). (1,15) The MOB is not a component based System, but a set of rules concerning dimensions, a set of ranges of preferred components and a set of standard details. It is designed to deal with a wide range of building types. Cedar 3 is intended for use at the sketch design stage, and its primary aim is to facilitate the comparison of alternative building geometries and site layouts with respect to capital and running costs. It provides facilities for building description input and editing, cost analysis, elevator selection, thermal analysis, daylight calculations and energy cost calculations.

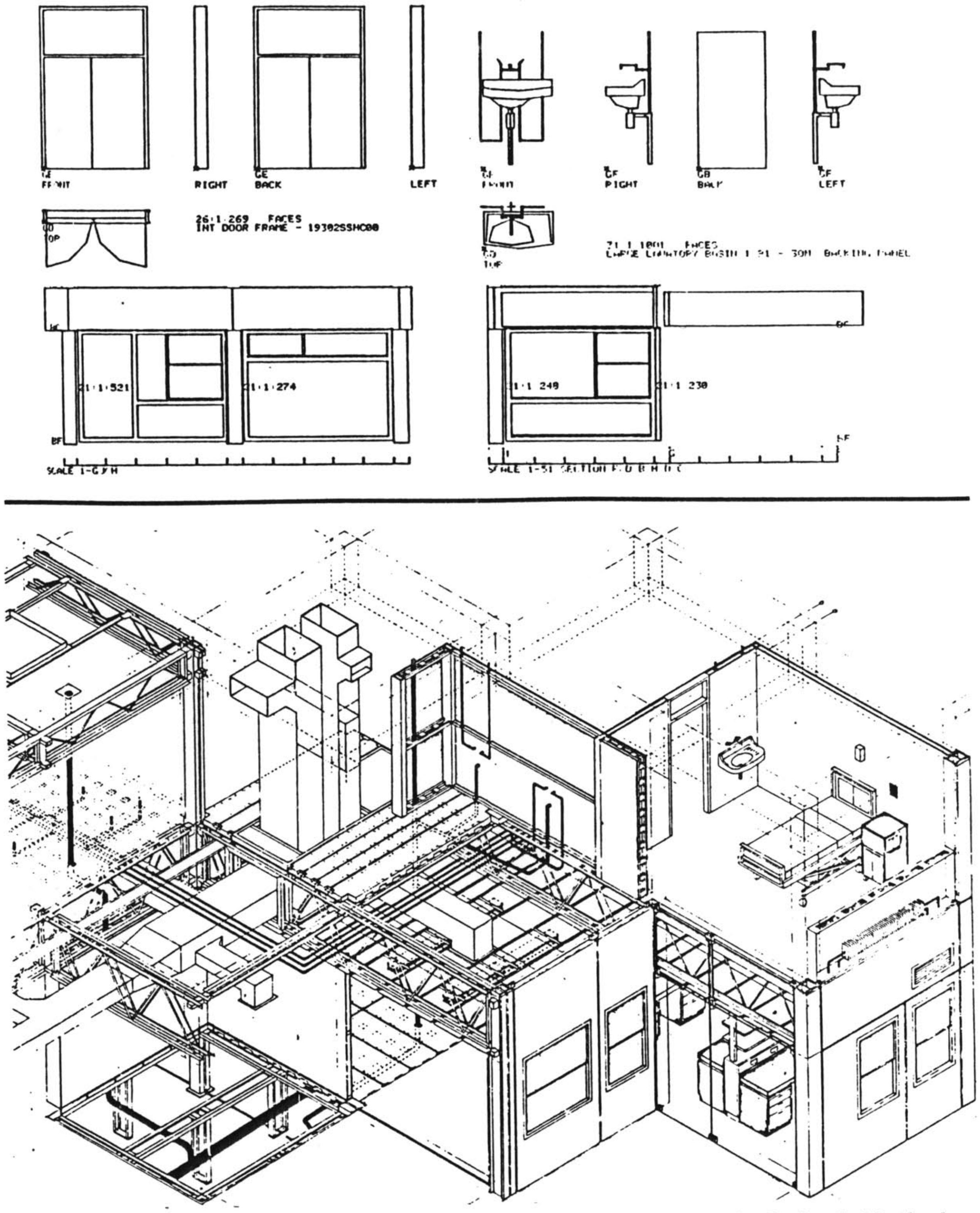




Top : Automated floor slab, cladding, and rooflight detailing by OXSYS. (a) automated location of floor slab elements (given beam and column locations) (b) Automated filling in of cladding panels on elevation. (given plan and section) Source : Mitchell 1977 p 244. **Bottom : OXSYS (a) Elevation tartan grid. (b) Complete building description with components located in tartan.** Source : Mitchell 1977, p 232



Several alternative designs generated using OXSYS, for a Hospital in the Oxford Method. Source William Mitchel p 106



Top : Drawings produced by OXSYS form the CODEX of Oxford Method components. Source : William Mitchell (1977) p 231 **Bottom ORHA OXFORD METHOD** Source : Bijl, 1979, Integrated CAAD Systems. p3-22.

The KORAB system. (12)

KORAB is a computer-aided building design, appraisal and documentation system dedicated to the design of residential buildings. The specifications for the system were developed in Poland in 1976 and software writing began in 1977. The KORAB system consists of 4 subsystems, namely the "Flat" subsystem, the "Building Section" subsystem, the "Building" subsystem and the "Prefabrication Plant" subsystem. Input information to the KORAB system consists of sketch designs of flats, building sections and buildings of the housing estate project and the project realization plan. As an output the system produces the following information :

i) Evaluation of the feasibility of building designs from the component production standpoint.

ii) Documentation drawings and schedules of components for the preliminary design stage of the project.

3.2.4 References

1. Bijl Aart, Stone David, Rosenthal David, (1979) "Integrated C.A.A.D. Systems" p(4-1 - 4-2)

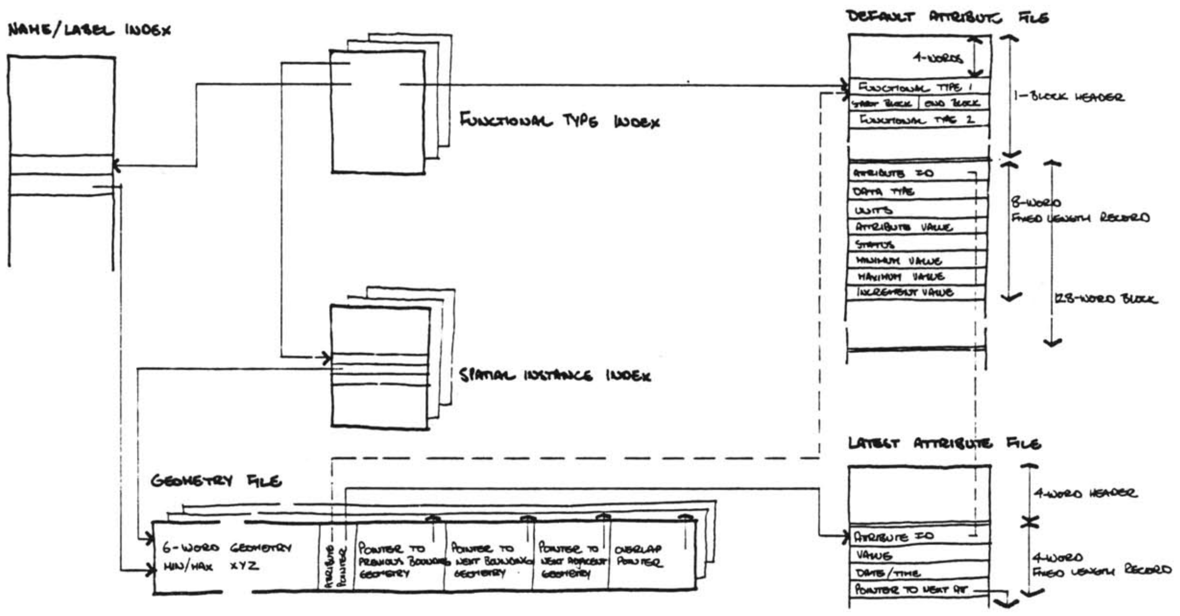
2. Chalmers J. (1972) "The Development of CEDAR" Proc. Int. Conf. on Computers in Architecture, York, 1972.

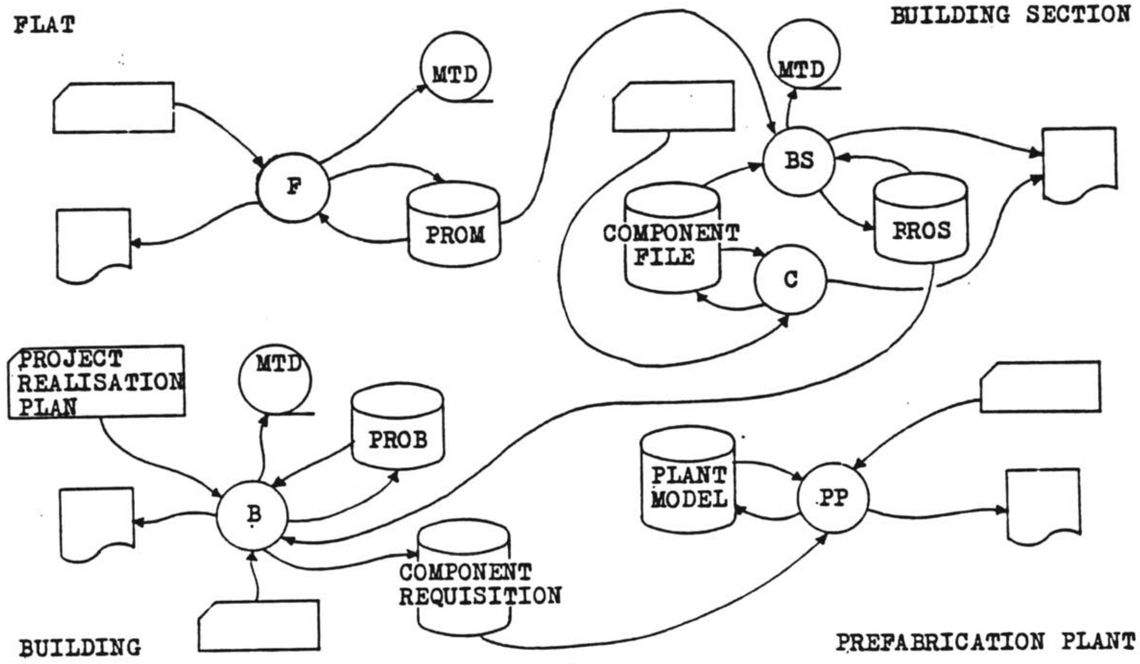
3. Chalmers J., Sampson P. & Webster g. (1974) "Data Structure Used in CEDAR" in Programming Techniques in CAD" Sabin. M. (ed) National Computer Centre.

4. Charlesworth D., Webster G., (1976), "CEDAR 3: Basic software for computer-aided design" CAD 76 Proceedings.

5. Cross N. (1977) "The Automated Architect" Pion Limited. London.

6. Daniel P.T. (1973) "A Plant and Buildings Draughting and Scheduling System," in J. Vlietstra and R.F. Wielinga (eds.), Computer Aided





Program packages: F - FLAT, BS - BUILDING SECTION, B - BUILDING, PP - PREFABRICATION PLANT, C - COMPONENTS. MTD - MT with drawings

CEDAR Building File Structure. Source : Bijl, 1979, Integrated CAAD Systems. p 3-16 **Bottom : An Overall Structure of the KORAB system.** Source : Kociolek A., Radwanski A., (1979) Computer Aided Design in Industrial Residential Construction.

Design. Amsterdam, Holland : North-Holland Publishing Co.

7. D' Arcy Richard (1979) "The Oxford System and its application." Proceedings of the International Conference on Computers in Architecture.

8. Hoskins E.M. (1972) "OXSYS: An integrated Computer_Aided Building System for Oxford Method. " Proceedings of the International Conference on Computers in Architecture. London: British Computer Society.

9. Hoskins E.M. (1973) "Computer aids for System Building" Industrialization Forum, Vol 4, no. 5 pp 27-42.

10. Hoskins E.M. "The OXSYS System" (1977) in Gero J. (Ed.) "Computer Applications in Architecture" Applied Science Pub.1977.

11. Hoskins E.M. and Bott, M.F. (1972), "DHSS : Development Planning System and Production Documentation System " Proceedings of the Intenational Conference on Computers in Architecture. London : British Computer Society.

12. Kociolek A., Radwanski A., (1979) ,"Computer-aided design in industrial residential construction." "Proceedings of the International Conference on Computers in Architecture."

13. Meager, M.A. (1972) "Computer Aids in Hospital Building" Proceedings of the international on Computers in Architecture. London : British Computer Society.

14. Meager, M.A. (1973) "The Application of Computer Aids to Hospital Building," in J.Vliestra and R.F. Wielinga (eds.), Computer-Aided Design. Amsterdam, Holland : North-Holland Publishing Co.

15. Mitchel William J. (1977) "Computer Aided Architectural Design" Van Nostrand Reinhold Company.

16. Patterson J.W. (1972) "Aids to improve the Designer's Control of Cost and Performance of Buildings, and to speed the Pre-Contract and

Post-Contract Processes," in W.J.Mitchell (ed.), Environmental Design:Research and Practice, Proceedings of the EDRA3/AR8 Conference. Los Angeles, California: School of Architecture and Urban Planning, UCLA.

17. Patterson J.W.(1974) "An Integrated CAD System for an Architect's Department." Computer Aided Design, vol.6 no. 1. (Jan. 1974), pp.25-31.

18. Purcell Patrick A. (1975) "Computer-Aided Architecture in the United Kingdom," in N.Negroponte (ed.), Computer Aids to Design and Architecture. New York : Petrocelli/Charter.

19. Ray-Jones A. (1968) "Computer Development in West Sussex." Architects' Journal, 12 February 1968, p.42.

20. Richens, P. (1974) "OXSYS: Computer-Aided Building for Oxford Method." Cambridge, England: Applied Research of Cambridge.

3.3 Recent developments : Artificial Intelligence and Knowledge Engineering.

3.3.1 Introduction

Computers have revolutionized most engineering and scientific fields, but their use has been mainly limited to numerical applications. Until recently, only human expertise in the area of well defined numerical algorithms has been coded to computer software.

The potential of using computers beyond strictly numerical applications is an important research issue. Presently human controls the flow of computer applications in an intelligent way. It is the user's ability to form judgements, make decisions, and assess alternatives, that defines the logical sequence of the execution of well defined numerical algorithms and that produces useful results.(7) In the past few years, however, there has been an increasing interest in artificial intelligence, which "... is the study of ideas which enable computers to do the things that make people seem intelligent", (11). Artificial intelligence has produced new means of presenting empirical judgement and in turn, performing logical reasoning.

Few fields have benefited from these applications mainly medicine, geology, and electronics. (4) However the potential applications in the building process are particularly stimulating and the mainstream of current research is directed towards this area.

3.3.2 Expert Systems

In a somewhat convoluted, but authoritative, attempt at a definition of Expert Systems, Addis81 (1) suggests:

'An Expert System is a means of capturing the knowledge of experts in the form of programs and data where disagreement among the experts are settled by mediation and the results refined so as to extract the essence of their knowledge in such a way that it can be used by less experienced people within the field. The usage of such a system can be monitored so that adjustments may be made semi-automatically under the guidance of the experts.The expert system is a tool and a means of coherent communication of the latest views of the experts to the users who may well be the experts themselves. The use of the system combined with a measure of importance provided by the experts gives a measure of the utility of what is being communicated. This recorded utility may then be used by a program to vet the knowledge so that the channel does not get clogged with redundant material.'

Feigenbaum81 gives a more accessible definition :(5)

'An Expert System is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant expertise for their solution. The knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of the expertise of the best practitioners in the field'.

In summary, however, and more informally, we can list the relevant characteristics of Expert Systems as follows. However it must be noted that not all existing ones display all of these features. (6)

i) They know a great deal about a limited but useful area of interest - Such knowledge being acquired possibly from experience but, more likely, from expert human tutors.

ii) They give advice conversationally in the manner of a consultant, and can understand and respond to simple questions posed in plain though perhaps specialized language.

iii) Their knowledge is embodied not in the form of conventional programs but frequently by means of separate modules containing sets of rules with corresponding actions. This feature makes for easier correction of deficiencies or errors in their knowledge-bases as well as the acquisition of new knowledge. Strictly, the implication of this is that the knowledge (facts and inference rules) exist independently of the program. This makes it possible (theoreticaly, at least) to use the same program with a variety of knowledge-bases able to deal with different tasks.

iv) Because the areas of interest they deal with are frequently ones where uncertainty prevails, Expert Systems often give their advice in probabilistic rather than absolute terms.

v) The questions posed By Expert Systems are limited to ones which are relevant to a particular line of reasoning. Thus if at any time the systems decide they have sufficient information to arrive at a conclusion, they do not continue to ask questions.

vi) Above all, Expert Systems can explain and justify their reasoning in such a way that experts can accept their credibility and non-experts can learn form them.

From the examination of several examples of already developed Expert systems it is possible to suggest that they perform best when the following conditions hold (6):

- Performance of the subject task is based more on factual knowledge than computational method.

- The area of interest is specialized and limited.
- It is possible to build-up a knowledge-base in a piecemeal fashion over time.
- The area of interest is one which experts exist and are available for consultancy sessions.

3.3.3 Case Studies of Successful Expert Systems

Since only a few Expert systems have been implemented and utilized in some way in the building process, in order to identify potential use of expert systems in this area it is important to seek expertise elsewhere. For this purpose a series of case studies of successful expert systems are to be presented.

Through this presentation, some of the applications of artificial intelligence techniques in expert systems related to various scientific fields are explicitly shown.

The applications to follow are both analytic and synthetic. In analytic applications the input data is the result of measurements or observations, and the expert system is employed to determine the cause. The sciences of the "artificial" are used to understand the "natural systems" (10). Such systems include diagnostic medical systems and oil exploration systems.

A synthetic system works in the reverse way. The objective is to create the artifact. The knowledge is used to synthesize the final product in order to fulfill the objectives that justify its creation. (The application of expert systems in the design of building systems is synthetic.)

CASNET : Model Based Method for Medical Decision-Making. The model is subdivided into three distinct parts: patient examination, pathophysiological states, and disease categorization. After examining the patient, the symptoms are then related to the corresponding states. Using that state and observed symptoms, a diagnosis is made along with lines of treatment of the disease. The process is clearly analytical, a casual associational network, and the database is relational, i.e. each node can have more than one parent , and the relations among the nodes are contained in sets of data, associated with the nodes.

MYCIN : Consultation System for the Physician. Mycin (9) is a goal-oriented algorithm, i.e., starting from the identified objective the algorithm reasons backward in order to establish a therapy for the patient. Rules are involved that either prompt the user to answer related questions or use already existing information in the system. THese rules will in turn invoke other rules which, depending on the symptoms, will be true or false. This will continue to until all relevant rules are exhausted, and a therapy is determined. The goal oriented approach to design therapy is is similar to that of design of a building system. The questions to inform on possible complications for the patient are parallel to the questions to clarify on the conditions in a particular site that the building will be built. Such a process will exclude one or more solution schemes for the structural configuration, and will propose a "best" choice.

INTERNIST : Diagnostic Problem Solving. Internist (8) is basically a knowledge based algorithm composed of two distinct components: disease entities and manifestation. Each disease has a list of symptoms and their frequency of occurence. Related to that is a database of diseases having as upper roots main categories of diseases which in turn are the roots of lower hierarchical specific categories. A list of alternative hypotheses are determined using the symptoms inputted by the user. Each hypothesis has a corresponding score. Using the hypothesis with the highest scores,a problem is compared prompting the program to select the program to relevant questions in order to further improve the diagnose. The diseases evoked will then be re-evaluated given the response until a definite conclusion is reached. This is an explicitly analytical procedure similar in concept to CASNET, but using a different approach.

PROSPECTOR : Consultant System for Mineral Exploration. Prospector (3) is a system intended to help geologist assess the potential of a site under consideration in containing a given mineral deposit. It is modelled after the actual reasoning of a an expert geologist when evaluating a given site. The system starts with an interactive session with the user in order to obtain information about rock formation and mineral deposits already detected. The data is then compared to models within the system of certain classes of deposits. Depending on the need, the system might ask for additional information before arriving to a definite conclusion. The system has been extremely helpful to oil exploration. Its similarity to applications for structural systems is even more pronounced by the proximity of the two fields.

DENDRAL : Inferring Chemical Structures. Dendral (2) is a heuristic search algorithm to list rational structures for organic molecules using sets of data from a mass spectometer and a nuclear magnetic resonance spectometer. Constraints, either user-defined or from packages, dictate guidelines within the system. The algorithm is subdivided into two parts: one to generate acceptable structures using the inputted data and the user defined constraints, and the other to deduce suitable molecular substructures using significant data patterns. The two parts are in continuous interaction thus guiding the system in producing the plausible set, containing the organic molecular structures that satisfy the previously defined constraints. The engineering program that creates the artifact of the molecular substructures is very similar to the basic concepts of systems building applications.

3.3.4 Potential Applications in the Construction Industry

There are several areas where Expert Systems can prove of particular utility in the construction industry. The design process is an area where simple quantitative rules are hard to be applied without oversimplification of the problems at hand. Furthermore the design process is surrounded by a number of associated scientific fields where again the nature and ambiguity of the issues involved as well as the difficulty of evaluating the importance of each factor at every instance requires a sophisticated tool that can provide consulting similar to that of a human expert.

The continually increasing number of different materials possible techniques as well as potential combinations of the above for the implementation of a particular project provide one more area where expert systems can prove particularly useful.

The nature of the the building process, where the project cannot be standardized beyond a certain extent due to the fact that it is tied permanently to the ground which causes the design process to be performed independently for every different project since the "givens" of the problem have to some extent changed. Since designing from scratch in each case is inefficient and out of the question, the need to make an intelligent and well justified match between the appropriate set of already made decisions represented by an already developed building system and the conditions of the particular design problem the architect faces, is one more hot area for the application of expert systems in the building process in general and the architectural practice in particular.

In all of these areas the utilization of expert systems proves particularly helpful for the following reasons : (6)

i) Once a knowledge-base has been deviced, Expert systems are easier to write, test and debug than conventional programs.

ii) In general, Expert Systems are easier to use and learn than all but the best conventional programs.

iii) In theory, at least, and to a marked extent in practice, the same program can be used with a variety of different knowledge-bases resulting in the easy setting-up of new Expert Systems for different subjects.

iv) Their self-justification feature is of special value in areas where the user must not only be presented with information but must also be convinced of its validity.

3.3.5 References

1. Addis T.R. (1981) "Definition of an Expert System" BCS Specialist Group on Expert Systems Newsletter No 3 Spring/Summer 1981.

2. Buchanan, B.G., and Feigenbaum, E..A., (1978), "DENDRAL and META-DENDRAL : Their applications Dimension", Artificial Intelligence, vol. 11, pp5-24.

3. Duda, R.O., Gaschnig, J., and Hart, P., (1979), "Model Design in the Prospector Consultant System for Mineral Exploration", Expert Systems and AI Applications, pp. 334-348.

4. Feigenbaum E.A. (1977) "The Art of Artificial Intelligence, Themes and Case Studies of Knowledge Engineering.", Fifth International Joint Conference on Artificial Intelligence, pp 1014-1029.

5. Feigenbaum E.A. (1981) Expert Systems in the 1980s in A. Bond's "Machine Intelligence".

6. Lansdown John (1982) "Expert Systems : Their Impact on the Construction Industry." Report to RIBA Conference Fund.

7. Pollalis Spiro N., Habib W.S. (1985) "Selection and Predimensioning of Structural Systems". Civil Engineering Department, Northeastern University.

8. Pople, H.E., (1977), "The formation of Composite Hypothesis in Diagnostic Problem Solving", Fifth International Conference on Artificial Intelligence, pp. 1030-1037.

9. Shortliffe, E.H. (1976), Computer-Based Medical Consultations : MYCIN, Americal Elsevier Publishing Co., New York.

10. Simon, A.H., (1982), The Science of the Artificial, MIT Press, Second Edition, Cambridge Massachusetts.

11. Winston, P.H., (1977), Artificial Intelligence, Addison- Wesley Publishing Company, Reading, Massachusetts.

SECTION 2. PROPOSAL :

A Computer Based Building System
within a General Computing Environment
for Architectural Practice

SECTION 2. PROPOSAL : A Computer Based Building System Within An Open Computer System for Architectural Practice

CONTENTS :

SECTION 2. PROPOSAL : A Computer Based Building System within a General Computing Environment for Architectural Practice

1. INTRODUCTION

Some of the implications of the application of the systems approach to building, were major reorganization of the building process, vertical integration through the building design and production activities etc. These changes in the building process were in several cases introduced with a considerable degree of success but in other cases could not be realized without considerable large scale institutional changes. Furthermore they proved to be more suitable for some market conditions and for the development of a limited portion of the built environment.

Systems building has proved to be particularly suitable for applications where the size of the individual project is increased and there is potential for a greater degree of standardization, facts which result in considerable savings in resources due to economies of scale. However it is a primary underlying assumption throughout this document that systems building can prove particularly useful when applied in a wider range of projects than those where mass production and large scale standardization can be easily achieved. If this is succeeded, not only will a much larger portion of the built environment be produced more efficiently but also the application of the systems approach to building will be dissassociated from a particular type of built environment which is repetitive and standardized.

Although large scale standardization is in several cases extremely helpful and economical, there are several types of buildings where due to their function as well as the context within which they are built and bound to operate, cannot tolerate or afford uniformity in their design. These buildings which still today comprise a very large part of the built environment produced are currently being built conventionally, a fact which results in a considerable waste of resources. This waste can only be eliminated if standardization is achieved at a smaller scale allowing the architectural firms which are commissioned to design these projects to effortlessly select building systems and subsystems from an open market and incorporate them into the building regardless of its scale.

Therefore this work is based on the primary assumption that in order for systems building to be the rule rather than the exception in the production of the built environment and for its full potential to be utilized, it is important for some of the facets of system building as the incorporation of building systems and subsystems in the built environment, to be generically utilized in a much larger scale than it is today, and for a much wider range of projects.

In order for this to happen a more open approach has to be possible and the appropriate facilities i.e. access to information, knowledge about the Building System as well as tools that facilitate its utilization, have to reach each and every architectural firm, regardless of its size. Every designer of responsible individual for a particular building project must be able to have this access to information and must be provided with a wide range of building systems and subsystems for selection from the open market.

This requires considerable reorganization of the whole building process which implies innovations that will cut a swathe across many of the fragments of this industry. This is made posible by the new facilities provided by the digital computer and more specifically in our days by the new generation of low cost hardware which suits well the needs of the fragmented building industry.

Based on the above this thesis aspires to investigate the possibility of efficient application of Systems Building in every day architectural practice. It focuses in a scenario which promotes the utilization of industrially produced building systems and subsystems in the building process and the resulting incorporation of such systems in the built environment, regardless of the size or nature of the project.

Therefore this work not only proposes a scenario for the development and the use of an improved building system that is appropriately utilizing the advantages of digital computers which guarantee better handling and manipulation of information and much more efficient interface with the user, but also concentrates on a scenario for the operation of an architectural firm that is realistic with the current developments in both hardware and software, can work regardless of the size of the firm and enables the utilization of such an improved building system.

In this chapter of this document the theoretical foundations for the application of systems building in architecture along these lines are explicitly stated. Two new notions are introduced in this process : The first is that of a Computer Based Building System, which is the the improved product of the evolution of the conventional "Paper Based" Building System so as to meet the new standards in the manipulation and transfer of information. The second is that of an Open Computer System for Architectural Practice which is conceived as the appropriate computing environment for the operation of the architectural firm and the generic application of Systems Building techniques in architectural practice, and is a prerequisite for the utilization of a Computer Based Building System.

These two notions i.e. the C.B.B.S. and the O.C.S.A.P. are the two interlinked facilities, components of a computing environment for architectural practice, which enable the generic application of systems building in architecture along the assumptions previously made in this document.

The C.B.B.S. is the product of the parallel development of a building system and a computer based information system which contains the rules of the system and is responsible for the interface with the user of the system, the project architect. The C.B.B.S. is developed by the building system or subsystem manufacturer contains information dedicated to the specific method of building and can be use for advertising purposes as well as for the increase of efficiency in the design process when utilizing the specific system.

The O.C.S.A.P. is a modular and integrated computer system resident within the architectural firm, which provides enough flexibility so as to be able to be easily accommodated for the utilization of a variety of C.B.B.S.'s in the form of overlays, even within a specific project and with no effort or specific retraining required by the project architect.

This chapter of the thesis first introduces the notion of the C.B.B.S. assuming previous familiarity of the reader with the notions presented in the previous chapter. Then it introduces and describes the O.C.S.A.P. and finally focuses on a scenario for the use of the C.B.B.S. within the O.C.S.A.P.

2. COMPUTER BASED BUILDING SYSTEMS

2.1 The Notion of a Computer Based Building System

2.1.1 Computer Based Building System Versus Paper Based Building System

William Mitchell started his presentation on the hot topic "What is Computer-Aided Design" on the ACSA Administrators Conference in Washington, on December 13, 1983 by stating that when automobiles first appeared they were called "horseless carriages" and by predicting that in a few years, simply by taking it for granted that we use computers to support the design process, the term "Computer - Aided Design" will sound just us strange to us.

Regardless of the controversial arguments that such a statement brings among experts in the field of CAD and practicing architects, regarding the extent to which this computer-aid will be provided and the kind of problems it will help to address, (as well as the new problems that it is likely to create) I shall at this point attempt to use similar argument in the case in point namely the Computer - Based building system.

The state of the art of Building systems development today, does not take advantage of the potential offered by current developments in computer technology, to the extent that the final product, the building system still remains in most instances practically unaltered despite the undergoing revolution in information technology.

This fact urges the need for a new term, "COMPUTER BASED BUILDING SYSTEM" to be invented in order to differentiate the new product, result of the interface of systems building with a computer based information system that handles and conveys information about the rules of the system, with a "conventional", paper based, building system.

2.1.2 Computer Based Building System Versus Computer Aided Building System

Within the "Computer Based Building System" the term "BASED" is not accidental either. It implies that since the computer based information system is an integral part of the whole Computer Based Building System, as the case with every system, design considerations of the part should affect the design of the whole and therefore all parts of the system should be designed simultaneously in order for maximum efficiency to be obtained. Therefore the word "based" as opposed to the word "aided" differentiates the proposed system from one where the design of the computer system has followed the design of the original building system.

The prediction here is that since many building systems or subsystems have been already developed and are already available in the market, we shall be seeing several modifications of existing systems into "Computer AIDED Building Systems" for some time to come.

However, this document introduces and describes the new concept of a Computer Based Building System and advocates the advantages that this proposed system potentially has over the several already developed "computer aided" building systems given the range of possibilities available by the new technology. The model of the Computer BASED Building System, should and will eventually prevail in the design of new building systems since it results in a much more coherent and efficiently designed whole.

2.1.3 Potential of a Computer Based Building System

The application of Systems building in a particular project has several advantages related primarily to savings in time and efficiency of the artifact - the building - which have been explicitly stated in the first chapter of this document.

These advantages can be intensified more by the interface of systems building with powerful computer tools and the integration of the facilities into a coherent whole, a Computer Based Building System.

Systems building is involved with extensive study of the various individuals involved in the building process, their often conflicting interests and goals as well as the level of coordination of their activities towards a common goal for a faster, and more cost-efficient construction process, combined with an improved final product. The building system with its new format will be able to convey the necessary information to the various participants of the building process much more efficiently that the paper based building system. A well planned parallel utilization and proper integration of computer facilities within this process significantly facilitates the role of the these individuals.

The project architect is ideally initially provided with a decision support system, that helps him select the appropriate building systems or subsystems to be utilized in a particular project. Then, within a familiar already computerized working environment, he utilizes a computer based building system which conveys to him through an interactive process and with minimum effort the rules of the building system, and prevents him from performing dangerous deviations from these rules. This easy access to information saves the project architect a lot of time, and ensures a much more efficient and consistent design.

The building systems designer has now powerful tools for accessing and maintaining information about systemic elements, and is entitled to design computer based building systems, i.e. design a concomitantly a building system and a computer based information system that will contain and convey the rules of the building system. His job is facilitated by several application packages or at best from the utilization of an integrated system that is specifically designed to support systems design.

The product manufacturer has a tool that due to increased ease of use by the project architect and is therefore more marketable. Furthermore he has the option to reduce production costs by utilizing the developments in the area of Computer aided manufacture in order to standardize and automate production while ensuring quality that meets a given set of performance specifications without need for testing the individual product.

However before this scenario can be realized, and Computer Based Building systems can prove themselves as useful developments that facilitate the building process and provide the missing link that will enable the effortless utilization of industrialized building systems in everyday architectural practice, several preconditions have to exist.

These are primarily related to the environment within which the C.B.B.S. is to be used which has to mature in order to make possible the efficient utilization of the new tool. Several major commitments have to be made both by individual entrepreneurs and most important by public authorities. In the next part of this document these preconditions shall be examined in detail.

2.2 Preconditions for the Development of a Computer Based Building System

2.2.1 Preconditions Related to the Design Process to be Followed

2.2.1.1 Utilizing the Systems Approach

In order for the C.B.B.S. to be efficiently designed a systemic process has to be utilized. This implies a sequence of activities as follows :

- Development of a set of user requirements for the specific C.B.B.S. - Development of a set of performance criteria and evaluation (testing) procedures i.e. a set of performance specifications. - Design of the system so as to meet these specifications. - Evaluation of the system against performance specifications. - Feedback. - Repetition of Last three steps until performance specifications are finally met.

Each one of these activities - steps in the design process - will be elaborated below in order to clarify their role in this process.

A generic set of user requirements related to a C.B.B.S., has already been developed. It is realistic, based on the current state of the art of the technology and in accordance with the role of the C.B.B.S. and the needs of the project architect using a C.B.B.S..

However, Building Systems, contexts and needs of potential users are likely to vary considerably from one C.B.B.S. to another. This makes it imperative, before indulging in the long and resource consuming task of C.B.B.S. development or develop a specific set of User requirements directly applicable to the project at hand.

This does not imply that the design team is likely to start from scratch but merely that the generic set of existing user requirements are likely to be "remoulded" considerably for the particular occasion.

The issues of concern the team at this stage are potential users, material and size of components, number of subsystems involved, openess of system and potential interfaces with other subsystems, probable size of project units to be produced, etc.

Once the user requirements have been established, it is important to ensure that the system to be designed will satisfy them. In order to achieve this goal it is important to develop a set of performance specifications for the product to be designed before the design process begins.

The first step in this process is to develop a set of performance criteria, that the product must meet. Once this is done it is essential to establish a set of explicit evaluation procedures in order to determine whether each one of these requirements is met.

The performance specifications for the C.B.B.S. involve issues related to the properties and interfaces of the physical components of the building system, the management of the system, as well as the design and structure of the computer based information system that handles the interface with the user of the C.B.B.S..

The next stage involves the process of designing a Computer Based Building System i.e. a Building System interfaced with a computer based information system, in order for the performance specifications to be met.

The design team has to concentrate on a wide variety of issues related to conventional building systems design and at the same time has to be concerned with the structure and the interface of the computer based information system with the building system.

The designed product has to satisfy the set or user requirements that were set for the specific C.B.B.S.. Therefore it must be tested continually against performance specifications to ensure consistency with the performance criteria. The design of a C.B.B.S. as with every Building System is a process of trial and error and finally leads to a product that successfully stands up to the established evaluation routines therefore guarantees user satisfaction and efficient performance.

The cycle Design - Evaluation against routines, - Feedback, is repeated several times until this goal is achieved. When this has taken place, it is time for involved entrepreneurs to make some important decisions before any major financial commitment related to industrial production and marketing of the system is undertaken.

2.2.1.2 Parallel Development of Computer System and Building System

According to systems theory, within a system the design of every component of the system, should be influenced by the required performance of the whole system.

The physical components of a building system as well as the Computer Based Information System that contains the rules for their assembly, are obviously part of the same system, both developed in order to serve the interests of the product (B.S.) manufacturer, and to operate efficiently within the same environment.

This is the first argument that in theory supports the need for parallel development of the two components. If one of them - The Building System - is developed first, some important performance requirement for the operation of the C.B.I.S. would most likely not be considered at this stage and this is likely to lead to a non optimal solution for the whole system.

A good example of this relates to the varying user friendliness of the C.B.I.S. resulting from the modular coordination imposed by the already developed building system. It is often the case that a properly functioning already developed building system proves very inappropriate for the design of a efficient and user friendly computer system that conveys information to the user, most likely due to the nature of the modular coordination among subsystems of the building system. Minor details in the design of the building system that could probably have been resolved differently if the issue was raised at the right time, may imply that the user is forced to extensively zoom in and out of the drawing or that the facility that overcomes this last problem is slow or memory consuming.

A very important argument advocating for the parallel development however is the extent to which the development of a C.B.I.S. leads to useful observations for the building system, which can lead to improvement of the system to be manufactured before it starts to be industrially produced.

By building simultaneously the C.B.I.S. that contains the rules of the building system, it is easy to remain conscious of the implications of design decisions at each step of the design process. The design team is forced to a discipline that is imposed by the computer, which is particularly suitable for the development of a building system. Each design rule once decided upon has very clear implications which are apparent from an early stage and in case it is changed the implications of this change are also immediately apparent.

Thus at every step of the process the members of the design team, are particularly conscious of the rules of the system as well as their implications, the degree of compatibility of subsystems as well as potential existing inconsistencies in sizes and interfaces among components. Furthermore the design team has at every step a clear idea about the clarity of the rules of the system as well as the ease with which these rules can be transmitted to the user. One of the performance requirements of the Computer Based Building System, relates to the ease with which the user of the system can learn about the system or use it to perform certain tasks (e.g. design a building).

The scenario for the parallel development of the components of a Building System and a Computer Based Information System that produces the desired Computer Based Building System, implies that a single design team is responsible for the development of the whole C.B.B.S.

This is particularly helpful since it implies that there is no need for a second design team to become familiar with a building system designed previously by a different team as well as the underlying design principles for its design (with which it may agree or disagree), in order to produce a Computer Based Information System that will convey the rules to the user. This obviously implies savings of resources as well as increased familiarity of the people that are involved in the development of the C.B.I.S. with the rules and design considerations of the Building System.

On the other hand this also guarantees that the members of the design team that will be trying to establish the set of systemic rules that would lead to a building system which will best meat the user requirements and the already developed set of performance specifications, will be working on the computer and therefore have a wide range of computer facilities to assist them in their task. These may be simply ease of access to a large pool of information, facilities for simulation, etc.

2.2.2 Preconditions Related to the Environment of the System

2.2.2.1 Identifying the User and the Environment Within Which the System is to to be Utilized

According to systems theory the first step when attempting to study a system is to define its boundaries. This implies that one has to clearly distinguish between the environment of the system, i.e. issues that determine the operation of the system without being affected by the system and are therefore given constraints in the design of the system, and the resources of the system which lie within its boundaries.

In any building system one may identify several such "givens" lying outside its environment. These include market conditions, available raw materials and resources in general, legal framework etc. These issues have been extensively addressed in literature in the past. Here we shall only deal with the extra environmental constraints imposed in s C.B.B.S. that were not considered in a "conventional" Building System.

Such constraints are imposed by the environment of the Computer Based Information System that is the new component that is added to the Building System in order for the C.B.B.S. to be produced. This environment is defined by the user of the computer based Building system whose convenience is the primary goal behind the design concept of a C.B.I.S. as a necessary component of the C.B.B.S..

The user of the C.B.B.S. is the person who either selects the particular building system from a list of possible alternatives because he finds it suitable to address efficiently the particular problems associated with the specific project he is commissioned to work on, or is simply asked from some authority to design a particular building, or project by utilizing a given building system or set of compatible subsystems.

In most cases this person is likely to be an architect. In some environments it can also be a small developer or contractor, especially in smaller projects. However in this document we shall assume that for most projects of some size that utilize systems building techniques, this person is going to be the project architect and for reasons of clarity as such he is going to be referred to in the text.

At this point it is important to explicitly state some assumptions which are made by the author based on his own motivation and interests and which serve as a basis for the further definition of the environment within which the Computer Based Building System is, and should be, used.

The first assumption made is that it is of importance to envision a building process within which industrialized building systems or subsystems can be utilized in the majority of buildings designed and built.

The next assumption is that the application of systems building to architecture must not be limited to projects of a specific type, whether this implies scale, cost per square foot or scale and type of authority that is in charge of the project.

An other assumption made at this point is that it is particularly relevant and important to enable the applications of systems building to architecture in buildings of small scale, which are developed individually in the private sector, where the negative implications of mass producing built environment do not apply.

Finally it is assumed that within the context defined by these assumptions it is particularly relevant to promote an open system approach which will allow the user, small developer or project architect to select building subsystems from a catalogue and incorporate them in a particular building.

Based on these assumptions which indicate the writer's particular concerns and interests in the application of systems building in architecture, one can attempt to define the environment within which the Computer Based Building System shall have to efficiently operate.

It is clear that one does not wish to restrict the potential for efficient utilization of systems building to a few relatively large, dedicated architectural firms but, on the contrary, would attempt to make them easily applicable to design within the majority of architectural firms.

Thus, the computer based Building system is designed to be used by the project architect operating within an "average" firm and therefore, the operation of such a firm is of relevance for a scenario for the use of the system.

2.2.2.2 Ensuring that This Environment Enables the Efficient Utilization of the System

In order to study the environment within which a C.B.B.S. is likely to be used based on the assumptions previously made, it is important at first to examine its immediate environment within the firm as well as facilities outside the firm that might potentially be available.

The C.B.B.S. no matter how sophisticated, for very practical reasons cannot be utilized within a firm which is not computerized and does not utilize a flexible computer system which will allow the "overlay" of the C.B.B.S. upon the existing hardware and software configuration.

Furthermore, at least for the present and until it becomes the rule rather than the exception that the building process is based on systems building principles and commences with systems building selection, the environment of work of the project architect will have to be considered a given in the design of the C.B.I.S..

The system must be designed in such a way so as to suit the needs of the project architect without causing by its utilization deviation from his customary process of work within his firm. On the contrary it must make the life of the architect easier so that there will be increased incentive for him to wish to use the particular C.B.I.S. system and therefore the Building System. The C.B.I.S. by its user-friendliness is supposed to increase the market potential of the whole C.B.B.S. as opposed to a conventional Building System documented within a manual.

It is thus helpful for the development of a scenario for the use of the computer based Building System, to give some consideration to the overall computer facilities of the firm within which it is likely to be utilized. The nature of these facilities (i.e particular hardware and software configuration as well degree to which this is utilized) depends to a large extent upon the size of the firm, the degree of access to information outside the firm through network facilities or modems, the individual design process preferred within the firm as well as level of computer literacy of the users.

Thus the next part of this document will contain a detailed description of an Open Computer System for Architectural Practice which, although totally independent from the C.B.B.S., with the assumptions made in this thesis, forms part of its environment and is the most important precondition for its use and therefore for its development.

3. THE NOTION OF AN OPEN COMPUTER SYSTEM FOR ARCHITECTURAL PRACTICE - ENVIRONMENT FOR THE USE OF A COMPUTER BASED BUILDING SYSTEM

3.1 Potential of Microcomputers

The forth generation in the evolution of computer technology, which started in 1971 - 72, was characterized by the interconnection of numerous computers to form a large-scale computer network, and the introduction of relatively small and inexpensive minicomputers and microcomputers.

A major factor which led to this new generation of hardware was the development and refinement of techniques for manufacturing miniature electronic silicon circuits. As William Mitchell characteristically stated back in 1977 (Mitchel 77) Microprocessor technology had reached the point "where tens of thousands of electronic components can be fabricated upon a chip less than a quarter of an inch square at a cost of a fraction of a cent per component".

The development of small and inexpensive hardware as well as the capability to network several machine together has increased dramatically the potential uses of computers and their importance in our lives. The design of desk-top computers has enabled access to computer facilities from a wide variety of environments, utilization of the tools in professional practice, education and as integral parts of most functional devices in our everyday life.

The resulting increase in the size of the market together with the decrease in the size of the individual hardware components and the facilitation of communication between these components, indicated a shift towards a more open configuration in the structure of the system.

Software too, has been developed along the same principles, with emphasis on the links and interfaces among various programs as well as their compatibility with a wide range of hardware.

This move from the turn key closed system approach towards a more open modular configuration of computer systems, has also significantly influenced the building industry. It allowed for the first time access to computer systems to many small firms that provide professional services.

Architectural firms are clearly among those firms that have incentive to seriously consider the significance of such developments. A wide variety of inexpensive software has already been developed in the areas of Digital Modeling and Data Processing Analysses. Also recent developments have allowed the development of a Visual Information System incorporating innovative developments in video disk technology.

So, the recent rapid development of microcomputer technology with the resulting expansion of the market for computer applications that run on desk-top computers and the parallel development of software along these principles, has initiated a switch towards more open computer systems. This principle is of particular relevance for computer systems developed for architectural practice.

Author, operating an Enhanced Microcomputer workstation, equipped with high resolution graphics, digitized and / or mouse as well as a Video Disk Player and Screen.

3.2 The notion of an Open Computer System for Architectural Practice (OCSAP)

In the first chapter of this document, CAAD systems have been extensively described, as tools that support the project architect in the design process. However each system seemed to be appropriate for design within a different environment. Some were designed to be used for component building, some others for rationalized traditional construction, etc.

The other important specification for a such systems that has been previously stressed is the extent to which these systems were integrated. By this was implied the fact that the user had to enter data only once, and the system maintained consistency throughout the process.

The new concept that is introduced now, is that of modularity of a given computer system. It is the reflection of an attempt to move towards a more open system, which has been triggered by the relatively recent increase in the importance of desk-top computers.

The system proposed is open, both in terms of its hardware configuration and, most important, its software configuration. It consists of small functional units that can be easily added or subtracted from a given configuration without changing essentially the nature of the system. Furthermore most of these units are likely to be relatively cheap, off-the-shelf packages.

Although this model is becoming increasingly common in several environments that offer computer facilities for bussiness applications, it is likely to take a slightly different form in the area of Design and Architectural Design and Practice in particular.

Within the environment of an architectural firm, the complexity as well as variety of the tasks the computer system is asked to perform, as well as the large data-bases of drawings that have to be maintained call for a relatively complex highly sophisticated system. This system in order to be used efficiently requires a longer period for training of its users as well as some degree of customization according to the specific needs of the firm.

Once this investment in time and capital has taken place there is an understandable reluctancy to move onto a different system unless very obvious reasoning supports such a decision. More so it is not possible to utilize concurrently two separate integrated computer systems and have each one perform those tasks for which it is particularly suitable, since this would result in a non-integrated overall system, as well as serious problems with the management of graphical data and with training personnel for its use.

The varying workload so typical of an architectural firm is definitely one argument in favor of a more flexible open system. However the varying potential uses in which each architect expects his system to perform efficiently as well as the wide variety of definitions of efficiency for each case that are advocated by individual architects is the primary argument that supports this model of an open system.

The proposed Open Computer System for Architectural Practice, (OCSAP) is a demonstration of a computer system built along these principles. It is a modular system both in terms of hardware configuration but most important in terms of software development. It assumes that there are several parts within the system which can be interchanged since each one has generic interfaces that allow it to communicate with the outside world, as well as appropriate hooks for use with a wide range of hardware.

The system is integrated and has facilities that are designed to provide services in the areas of digital Modeling, alphanumeric data processing as well as interactive storage and retrieval of visual information. This last facility shall be referred to as the Visual Information System.

Within all three of these generic facilities several levels of programming ensure the ease of customization of the software by the user, without preventing the software manufacturer to produce innovative updates of the generic features of the software. The system is thus always up to date without requiring any particular effort from the user for this purpose, who can invest his time on customizing it to his particular individual needs.

All three primary components of the system previously mentioned are essential for the efficient operation of the overall system. However the digital Modeling tool, the graphics editor, is not only the primary tool for input and output for geometric information, but it also the critical component of the system for the performance of complicated tasks related to design. Since the versatility of this tool eventually defines the utility of the system for particular applications and requires the longest training for its use, the graphics editor has a special place among the other components of the system. It is in fact the most permanent component, the one whose replacement shall cause the most confusion, and waste of resources for the firm. The model for the system therefore assumes a relatively difficult to change core of the system, the generic graphics editor, with increased capability of updating everything around it.

This central facility for digital modeling provides a set of generic operations for the manipulation of geometric data, the generic interfaces in order to communicate information consistently with the other parts of the system and the outside world, as well as a powerful facility for programming within its environment.

Around it stand a large number of potential other computer facilities, either as interfaced applications packages structured appropriately in order to form an efficient facility for alphanumeric data manipulation, as part of the visual information system or as overlays to the graphics editor, programmed with the facilities provided from the editor.

This approach clearly distinguishes two levels of software development and ensures within a stable environment a potentially infinite number of different uses of the graphics editor, each one suitable for a specific design task. However there is no need for particular training since the basic rules for the operation of the editor are the same in any instance and the hardware and software configuration are also familiar.

Several software vendors in the area of CAD have already made impressive progress towards this proposed area and have already developed graphics editors with several of the features that are required for this model to work efficiently. Also several other vendors, have already started to market software, programmed on the second level, within the facilities offered by these generic graphics editors.

3.3 The Modular Graphics Editor

3.3.1 The Graphics Editor - Introduction

Importance and Use

As already stressed the importance of the graphics editor within an open computer system for architectural practice, stems from its use as the primary input and output of geometric information, the kind of information the architect more than any other professional has to able to manipulate. Furthermore, the importance of this tool within an architectural firm is justified by the wide variety of its alternative uses, ranging from drafting to the most sophisticated applications in design.

The graphics editor might be used for component building, unrestricted architectural design, or simply to study the facade of a palladian villa.

Not only is the graphics editor expected to perform very differently within the various tasks involved in the design of any individual project, but different projects often impose their own restrictions and standards for the optimal performance of this tool. Still the design process as well as habits of the particular architect make the problem of defining a optimum function of a graphics editor even more complex.

Permanence of the Facility.

One other major consideration for the graphics editor is that it is a computer tool that once acquired by the firm, it is rather difficult to change. This is due to the considerable amount of initial investment, training and organizing that is required for this rather complicated tool to be set up and running and also, to the fact that once the system is running there is still considerable investment in customizing the editor with subroutines, libraries of parts, fonts etc., which make it more efficient. To this one must add the further complication of converting all archives of previous drawing files, to a format readable by the new graphics editor.

Therefore unless major breakthroughs are introduced from one particular company, it is rather unlikely that a specific firm shall decide to switch to a different software vendor for this critical central component of its software infrastructure since this would imply that this considerable amount of investment would be wasted. Although such switches are probable once every ten or fifteen years in order to make up for a bad choice of software vendor or unpredicted developments and highly efficient new technology, a considerable degree of faithfulness to already existing facilities is expected.

It is therefore of particular interest to the firm that it picks a reputable software vendor that markets a well structured program with potential for expansion.

Excluding Alternative Scenaria.

The illustration of the problems that a firm has to face in order to replace its old graphics editor with a new one, makes it unnecessary to stress the futility of attempting to utilize simultaneously more than one graphics editor in the firm.

Since the graphics editor has to perform efficiently in a variety of different situations, and the problem cannot be resolved by the parallel utilization of several facilities each efficient in the performance of some of these tasks, there is a need to conceive a single tool that performs the job.

However it is rather difficult to design, program, debug, maintain, and update a comprehensive graphics editor that performs optionally in all these different situations. Even if such a deed was undertaken and carried through efficiently the following major problems will most likely occur :

- The system will be too large and complex and will require a long time for users to get familiar with it, which is very bad, since it threatens to turn the process of becoming familiar with the tool into a very lengthy operation. The graphics editor has on the contrary to remain a handy, simple and easy to learn and use, tool.
- The system will be difficult to update, very rigid and very difficult to debug. It is also likely that if needed in the future, it will be highly unlikely that conceptual changes in the structure of the system will be performed with ease and without major changes in the use of the system and major inconvenience to the users. This implies that the "large and comprehensive" system will soon be far less efficient than available technology would permit.
- Most important of all, no matter how complicated the system becomes, the user, especially in the field of architecture and in the very controversial issue of design, will still find areas where the systems' performance does not suit his individual needs and preferences.

The point is made that a large comprehensive generic graphics editor, programmed, updated and maintained by a single software vendor, is both difficult to implement and unlikely to meet user requirements.

Instead, the answer to the problem can be provided by a different approach, that is advocated by an open computer system for architectural practice. This system has as its core a compact transparent graphics editor which provides only a primitive set of facilities for manipulation of geometric data, generic interfaces with the outside world, as well as a powerful programming environment for internal programming and further customization. Other interested software developers, including the user, of the system can develop overlays for this graphics editor and thus make it particularly suitable for specific applications.

3.3.2 Division of Tasks and Responsibilities at Various Levels of the Graphics Editor

TOP LEVEL - Permanent Core of the Graphics Editor.

A software vendor specializing in CAAD produces a generic tool which provides a set of finite drawing primitives essential for any application of the graphics editor. The package has to have the following features :

- Appropriate devices for the interface with a wide range of available hardware in the market.
- Appropriate hooks for interfaces with other programs.
- Appropriate facility allowing for customization within the graphics editor in the form of overlays.

The first issue related to the interfaces with other programs is quite obvious. Within an open environment, one cannot afford to invest resources i.e capital, but most important time, on a graphics editor that does not guarantee good communication with the rest of the world. The graphics editor will have to provide a facility for extraction of both geometric and alphanumeric information from a drawing file to the outside world in a generic (straight ascii) format, understood by other applications - components of the overall system. Also hopefully several easy to use facilities for extraction of attribute information that can be processed by a facility for alphanumeric data processing.

The second issue related to the customization of the generic editor is critical since on this feature depends whether the graphics editor will eventually satisfy the users needs, allowing for a sophisticated information tool to be built around it.

Responsibilities of the software supplier at the top level of the generic graphics editor:

The vendor provides a small package with the primitive geometric manipulations and geometric data base operations as well as the programming facility and the hooks with the outside world previously described. The data base management at this level has to be particularly sophisticated since inefficiencies at this level will have considerable impact on the performance of the whole system.

The software developer at this level is responsible for updating his facility according to new inventions and developments, correcting already existing inefficiencies if any as soon as possible. He also is responsible for providing facilities for efficient interface with every possible hardware device that may hook to the system and is available in the market. The vendor is also responsible to increase the efficiency of the programming environment within his program. He is also responsible for good documentation.

The major responsibility of the software developer however is to ensure that the changes and updates he implements in his recent versions of the software do not in any way affect work already performed by the user or other developers utilizing the programming environment within the graphics editor for further customization of the product for specific applications.

LOWER LEVEL - Programming within the programming environment of the graphics editor : In the development of the system there is however a second level of programming involved and it is at that that level where the graphics editor is provided with the power to be efficient in the various applications where it is likely to be utilized. This is the level where customization of the graphics editor occurs.

Thus the programming environment within the editor is utilized to create a large number of modules that are attached to the core of the editor. These modules can be conceived as overlays and consist of custom made menus, linked with specific files which contain sequences of commands that are invoked from the menus upon request from the user of the system.

Each one of these modules, is invoked separately by the user according to his immediate needs. No extra training is required for the use of these modules since the rules and syntax of the system is defined by the set of drawing primitives that are provided by the software sypplier. The system does not depend on any one of these modules in order to operate in its most primitive and suppressed version, however, any combination of such modules can be added to it in order to enhance it and make it more efficient in the performance of specific applications, as well as subtracted from it or replaced with some other module.

Conclusions

This model leaves us with a totally modular graphics editor. The tool is very flexible, can be easily updated and does not require extensive training. Furthermore the various modules can be programmed in two very different ways a fact which adds to the flexibility of the system. They can either be developed by independent software developers that are directing their activities at the task of customizing a generic graphics editor, or can be programmed by the architect himself.

The first case is most probable in terms of generic applications of a great magnitude, like developing a facility that makes the graphics editor efficient for entering site information, or for designing floorplans -as opposed to details- etc. Such modules have already started to appear on the market in the micro-world. This tendency is likely to increase dramatically once the marketplace gradually becomes more stable and is dominated by a few companies supplying graphics editor packages.

The architect is more likely, either by himself of with the help of programmers to develop much smaller facilities that nable him to format his drawings in a specific way or to change the sequence of several operations. He will even have the possibility of developing large and complicated overlays similar to those the middle level software developers are producing, a fact which will allow him to have facilities that are tailored to his specific needs. However it is expected and most likely that due to limitations of time and probable lack of interest by the architect, this will be the exception rather than the rule. It may also be assumed that in times of low bussiness a firm can invest on building up facilities that can pay off in the future a fact which was not possible in a non computerized firm or in a firm that invested a lot of capital in advance for a turn-key system.

This hierarchy in the building up of these graphics tools ensures efficiency while at the same time leaves a lot of room for expression of individual preferences and needs, a fact which is particularly helpful when the users of the system are practicing architects. It also decreases considerably waste of energy in maintaining the system. The individual programming at each level of this hierarchical programming environment is concerned with issues of different nature and has a clearly defined area within which he may exercise his power and control. The sphere of influence of the programmers at the top level does not coincide with that of the rest of the programmers working on overlays to the system.

This model guarantees within a rapidly changing micro-environment a graphics system that is both continually updated at various levels while at the same time has the stability in terms of basic rules of operation, as well as building up of facilities and graphic databases, that an architect requires.

This is the graphics editor on which the proposed CBIS is overlayed. Each CBIS for a CBBS must contain an overlay for the specific graphics editor that the architect uses. This means that when an architect wishes to design a building using a specific CBIS the only modification he has to make to his graphics editor with which he is already familiar, is to add to it one more module-overlay that makes his graphics editor appropriate for design buildings with the specific building system.

3.3.3 Essential Features of the Graphics Editor

Recommended features that need to be provided by the graphics editor will follow, presented according to the level of programming where they should ideally be considered. *Generic features of the system to be provided at the top- level of the graphics editor:*

- Maintenance of a data base of geometric entities like points, lines and polygons.
- Appropriate tools for interactive manipulation of these entities.
- Capability for naming groups of geometric entities as well as querying and manipulating efficiently the entities by means of their name.
- Capability for interactive display of these entities in several user specified formats i.e. Plan, Axonometric, Perspective, etc.
- Facilities for customization of the package i.e. internal programming, in a language that is appropriate, most likely some version of LISP.
- Generic hooks for interface with other programs.
- Both on screen and tablet menus for easier input.
- Capability for variable and easy output.
- Drivers for interface with a wide range of hardware and peripherals.

Generic features to be provided by programming at the lower level of the graphics editor. (OVERLAYS).

- Capability for the efficient manipulation of entities in an environment where few complex manipulations are frequently demanded.
- Interfaces with specific applications that are suited for work within a particular environment.
- Appropriate interface which allows the user to acquire expertise within a narrow area of interest.
- On screen documentation and explicit presentation of the rules and constraints imposed within a specific environment as for example a particular Building System.
- Performance of user defined procedures suitable for his own method of designing.
- Fast and efficient performance of several laborious user

defined drafting tasks that are frequently repeated within a particular firm.

Of the above facilities most are likely to be designed and implemented by independent software vendors. The last two are most likely to be developed by the user himself or by a programmer hired for this purpose by the user. However the overlay for the efficient use of the graphics editor when using a specific building system is designed as an integral part of the CBIS and is supplied to the project architect by the building system manufacturer together with a set of applications and modules that run with it as well as specific hardware that might not be part of the computer system of the firm.

3.4 Data Processing

3.4.1 Data Base Management

Any integrated computer system to be used in architectural practice has to develop a systematic method for symbolic description of a building, as well as a method of accessing and organizing the data that will be helpful in order for this description to be developed easily and efficiently. This includes the issue of organizing and storing both geometric and alphanumeric information related either to a particular building or to generic facilities, parts of the open computer system. There is also a clear distinction among physical and logical structure of the data.

The system accesses, and uses data that is required for the development of the various projects. This data is contained in the master file, it is used several times, and increased access to such data increases the power of the computer system of the firm.

This data can be geometrical, alphanumerical, or visual. Part of this data is likely to be directly linked to one of the overlays of the graphics editor, developed by independent software vendors and representing a considerable capital investment for the particular firm, or provided free from an interested individual, as in the case of a catalogue of systemic components provided from a manufacturer to the firm. Other can be propriatory, developed by the firm itself as part of custom made subroutines that were developed indoor. This part of the data is of considerable value for the firm probably more important for its operation than the actual software (program) and is likely to be kept indoors.

However there are examples of data and facilities, parts of the master file of generic information, that due to their size as well as the generic nature of their use, either cannot, or should not be kept indoor. Such data, as for example the data related to building regulations of a specific area, is not in any way propriatory and can only be maintained by a central organization. Therefore the architectural firm will have access to such data through the network, or at worse, a modem.

The other major set of data, that is utilized in an architectural firm is that associated with a specific project. This data is contained in the project file. If more than one project is currently in process, there are several active project files. The users of the system are manipulating the data primarily though input from the digital modeling tool, in order to develop a building description for the building to be designed.

Once this building description, containing both alphanumeric and geometric information, is developed, several data processing and analysis subroutines are utilized, in order to evaluate the efficiency of the building described. These analysis subroutines operate only on the data contained in the project file and most frequently in the alphanumeric data contained in this file. There is increased interest in the development of evaluation routines that require as input geometric data, but these require far more sophisticated techniques in the area of artificial intelligence and knowledge engineering.

Finally besides the currently active project files, there is also the historical archive of the firm. This is a data file where the most important of the data contained in the active project files is stored for future reference, once the projects are completed. This file contains primarily graphical information, i.e. vector drawings, and also visual information developed and used with the new and powerful video disk technology, related to the construction process, the final constructed project, etc. Obviously the requirements for speed of access for this last file are significantly reduced.

As already mentioned a distinction is made among physical and logical structure of the data. The physical structure of the data is concerned with the data storage media such as core memory, disks, magnetic tapes, optical disks, etc. It actually describes the pattern of data in computer memory and its representations by means of physical devices.

The logical structure refers to relationships between data, stored as sets of entities as e.g. files, records, fields, etc., which determine the different ways records can be accessed by using the facilities of a programming language or a data base management system. The logical structure requires relationships between attribute categories at different levels of hierarchies and subsets.

As an example a building can be conceived at a functional level as a set of functional entities or at the structural level as a set of components belonging to a set of compatible subsystems. In each case different techniques are required for the retrieval of a particular kind of information. Several such alternative generic logical structures for the representation of the geometric data associated with a particular building, exist. These were considered briefly in the first chapter of this document.

The Open Computer System for Architectural Practice would ideally provide the user with a generic graphics editor which can modify the data representation of the building description according to user specified requests that stem from the nature of the particular project. Since this is at the moment a particularly optimistic scenario, given the state of the art of existing digital Modeling facilities and CAAD systems, an alternative more modest scenario is proposed for the near future, so that it can provide an efficient environment for component building.

The digital modeling tool utilizes a polygon and polyhedron representation, which is suitable for conceiving the building at a structural level as a set of subcomponents. It is however less suitable for rationalized traditional construction, as well as building expertise onto the system, associated to design tasks and functional requirements of spaces.

The hardware configuration which allows the storage and retrieval of this large amount of information is also of interest at this point. It determines to a large extent how much information can be kept indoors, and how rapid one can expect the retrieval of this information to be.

The data of the system is stored primarily on magnetic media, and all but the historical archive of the firm has to be stored on some kind of harddisk that is on line. The historical archive can also be stored of detachable floppies, but most likely it is again going to be an external detachable harddisk. All harddisks utilized in the firm have to have tape-backup capabilities. To this scenario we have to add the potential of utilizing optical disks for storage and retrieval of data.

Merging videodisk technology and computer capabilities has made feasible an innovative data base management system combined with an information retrieval system which utilizes prestructured image disks that are controllable from a computer data base system. Using the optical disk as a storage media is a powerful innovation for several reasons. It provides high capacity of memory storage, independence of physical structure and logical structure, as well as capability of computer graphics and image display.

The major disadvantage of this storage system at present is that one can only write on the disk once. Although the cost of the disk is still very low related to its capacity for storage of information, it is likely that this system will find more applications in the storage of permanent information part of the master file and it is less likely that it will be utilized for storage of information that will require constant updates.

3.4.2 Alphanumeric Data-Processing - Analyses

In the context described in this document, data processing refers to any kind of application which performs an analysis using the alpha-numeric information associated with the graphic representation. It usually includes straightforward applications of the report generating facilities of the data base system.

Vector drawings created by the digital modeling tool contain alphanumeric information stored in the form of attributes within the drawing. These attributes can either be assigned to particular components that form part of the drawing or can be assigned to spaces that are formed by the insertion of the components. Different applications require a different approach on this issue.

Other alphanumeric information contained within vector drawings relates to the names given to groups of entities e.g. blocks or symbols defined within each drawing, which too are often useful for manipulation by the data processing facility, however most likely, harder to extract. The third kind of similar information contained in a drawing, text, is unlikely to be ever of some use outside the specific drawing within which it was created.

The data processing system is much more efficient a tool than the graphics editor for manipulating alphanumeric information. Therefore the strategy is to store within the drawing only a minimum amount of alphanumeric information. The data processing facility is then responsible for deducting from the minimal information provided from the drawing, the associated information and then perform several analyses and manipulations on this data. This strategy is particularly helpful in component building where the code of a component stored within the drawing can later be associated with a large amount of data.

A relational environment is conceptually most suitable for the manipulation of such data. However the link between the hierarchical environment of the graphics editor and a relational environment of a data processing system is not always easy to implement in a generic way.

The range of types of analyses that might be carried out at the various stages of a given architectural project are virtually unlimited and rather diverse. Since different tasks within a project are undertaken by appropriate specialists, each one contributes to some extent to the development of the final product, and according to each one's viewpoint, different types of analyses have to be carried out.

The list of potential analyses is endless, containing facilities for structural analysis, bills of quantities and cost estimates, real estate development, thermal, lighting, acoustic analyses, etc. The tendency now is to develop more facilities that can perform analyses that deal with non quantifiable aspects of the project, issues related more to architectural design, and a lot of relevant research is performed currently in the area of knowledge engineering and expert systems.

Several quite different environments exist, potentially appropriate for data pracessing each with its own strongpoints and shortcomings. One is clearly that of a data base that is suitable for storage of large amounts of data, easy retrieval and manipulation of the data as well as facilities for report generation. Within this scenario there are several alternative approaches, namely the hierarchical approach, the relational approach and the network approach. In most cases the relational approach seems to provide satisfying results.

An other alternative environment that proves more suitable for extensive manipulation of numerical data, is the spreadsheet environment. The spreadsheet environment is that of a large matrix. Each space - cell in this matrix contains information, or formulas for the manipulation of information. The simplest definition of a database stored in a spreadsheet environment is a list of information stored in a range of cells that spans at least one column and two rows.

The major generic distinction between a spreadsheet environment and that of a conventional data base is that the data base environment is more suitable for storing a large amount of data while the spreadsheet can perform more easily manipulations of this data. Therefore the spreadsheet environment is particularly helpful for dealing with analyses related to the building process, since most of them require a considerable amount of calculation and manipulation of numerical data.

It is therefore appropriate to utilize a data base management system for the storage of the large amount of data that is required at various stages of the design process and to use the spreadsheet environment for the manipulation of this data for specific analyses. Thus the relatively small amount of data required for a specific application is sent over to the spreadsheet from the data base where it is manipulated appropriately in order to produce the desired results.

A generic difference between the digital modeling facility and the data-processing facility is that the latter requires far less training for its use and does not have to deal with the delicate issue of manipulating graphical data. It is thus possible to envision a system with more than one environments for data-processing that are linked together and interfaced with the core of the graphics editor.

However each one of these environments as parts of the Open Computer System for Architectural Practice must allow for internal customization and development of subroutines that make it a versatile, user friendly tool. The various analyses performed within the spreadsheet environment are precisely that, overlays that are programmed within this environment by the use of appropriate macros.

3.5 The Visual Information System

3.5.1 Computer Graphics and Image Processing -

The Notion of the Interactive Data Image File

Computer graphics techniques are of particular importance to architectural practice. The are utilized in drafting, production of perspectives, sections, axonometrics or mapping.

A digital modeling tool, a graphics editor, is an efficient tool for the interactive generation and manipulation of graphics. This approach of electronic simulation, allows the architect to generate vector drawings representing perspective views of the building and manipulate them interactively by the use of a device like a light pen, a joystick, a mouse, or a digitizer of some kind.

However linking a videodisk to the digital computer, creates a new visual dimension which constitutes a valuable visualization tool during the design process. The system is used both in the final presentation and for the optimization of design decisions during the design process. It enables the architect to proceed with parallel electronic simulation and optical simulation. The latter enables the user to view an animation system prestored on the videodisk.

A closely related field of application of interest to the project architect is that of image processing, which implies computer analysis of visual materials such as photographs. Typical applications of this technology today is analysis of earth satellite photographs to extract data about land use, mineral deposits, etc. Such analyses can also be based on a graphics editor for interactive manipulation.

Additional use of a digital decoder will enable the system to process the images stored in videodisk. In this case vector drawings can be stored on the videodisk and downloaded to the local computer system for manipulation. Thus the building description of a given project, can be built by utilization of standard components which are stored as vector drawings in the videodisk. The computer's memory can be filled from the disk in any place in order to control specific sequences as viewer-movie interaction unfolds.

So a new new concept is presented as the basis for a data image management which combines image storage in videodisk either as slides and video sequences or as vector drawings, and image retrieval through a digital modeling tool, the graphics editor.

The utilization of video technology as an integral part of the open computer system for architectural practice, provides major improvements to the system, both during the interactive process of developing the computer resident building description of the building, but also during the development of the final output, the presentation documents.

Optical disk storage of information is particularly suitable for the storage of large quantities of information which is relatively permanent. Video disks are particularly useful for storage of visual information, as for example images of components included in Sweets catalogue, which is updated only once a year. It can also contain propriatory information for the firm, like exemplars of particular kinds of buildings, etc. The retrieval of this visual information is achieved through the graphics editor as well as some relevant analyses packages. The system is fully integrated. However for tasks related to the particular project, it is likely that common video tape technology is more appropriate. Here it is important to record information quickly related to the particular project, the site, etc., to be able to access it from the system without having to go through the lengthy process of pressing a videodisk, and to be able to put together video tapes for final presentation to clients. It seems that video tape technology is a particularly suitable visual information system to complement the graphics editor for manipulation and presentation of project specific data.

3.5.2 Principles of an Interactive Videodisk

The primary function of an interactive video system is random access of data. this is the ability to access any segment of a pre-recorded video program with speed and accuracy. The standard video disk commands include searching to a specified frame #, playing successive frames forward and backward at specified rates, freezing on a single frame, etc.

An optical (laser-based) video disk can store an enormous amount of information in a small space. It is possible to store more than 2 gigabytes of digital data on each side of the video disk which makes each video disk capable of storing approximately 54,000 images or the data contained in 5,000 floppy disks.

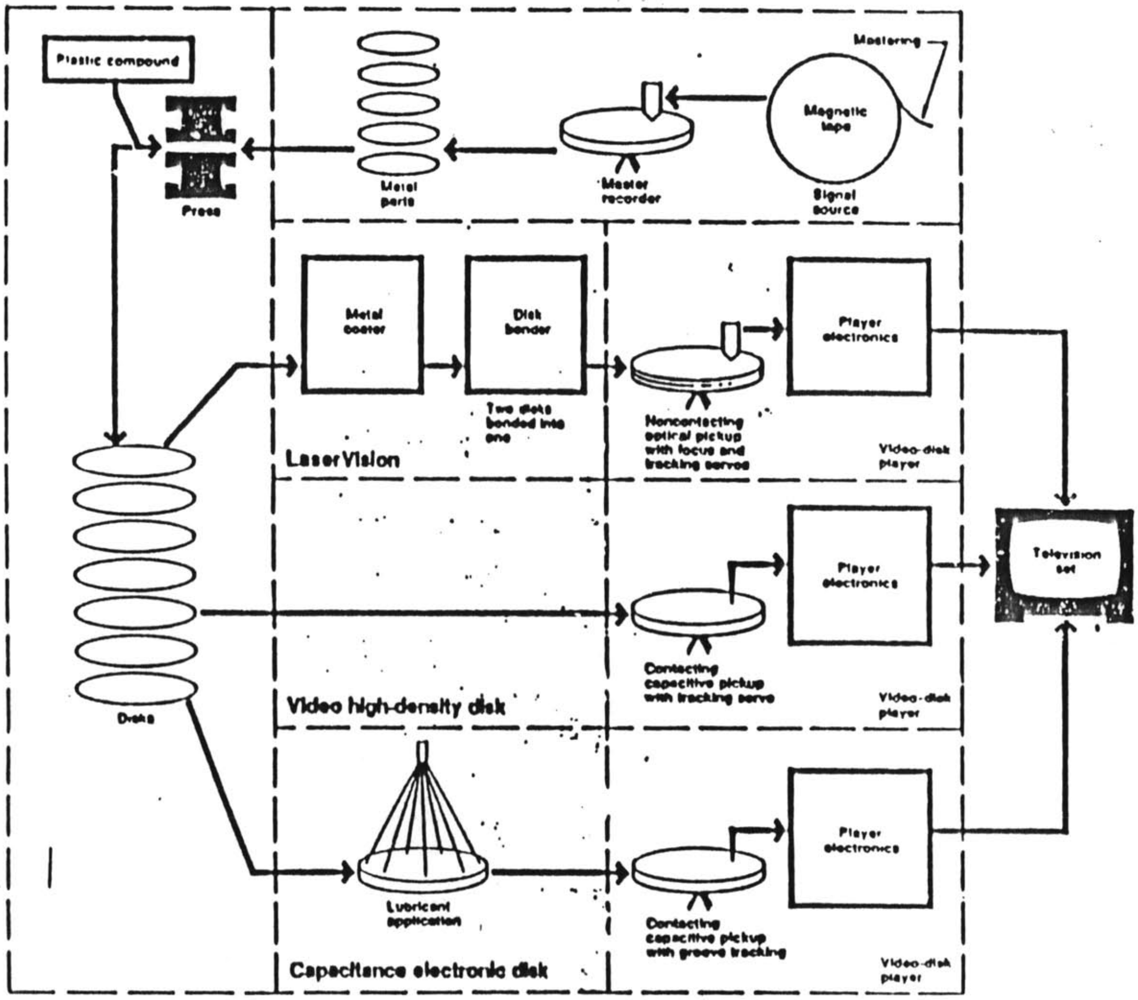
Its basic advantages besides high storage capacity are high speed random access, both motion and freeze-frame reproduction and two sound tracks. The videotape, with its clear single frame display, its fast access time, combined with low cost hardware and durability will provide unprecedented technical control and automation of instructional presentations, as well as presentations of ideas and schemes in professional practice. The proliferation of video-computer applications can be expected to include inexpensive portable devices and interactive TV networks.

3.6 Building Expertise into the System

Most analyses packages that form part of the data processing facility of the open system for architectural practice, tend to be particularly suitable for processing alphanumeric data and presenting the results of the evaluation to the user. This is understandable since it is easier for a digital computer to deal with quantifiable properties of the building and manipulate them efficiently.

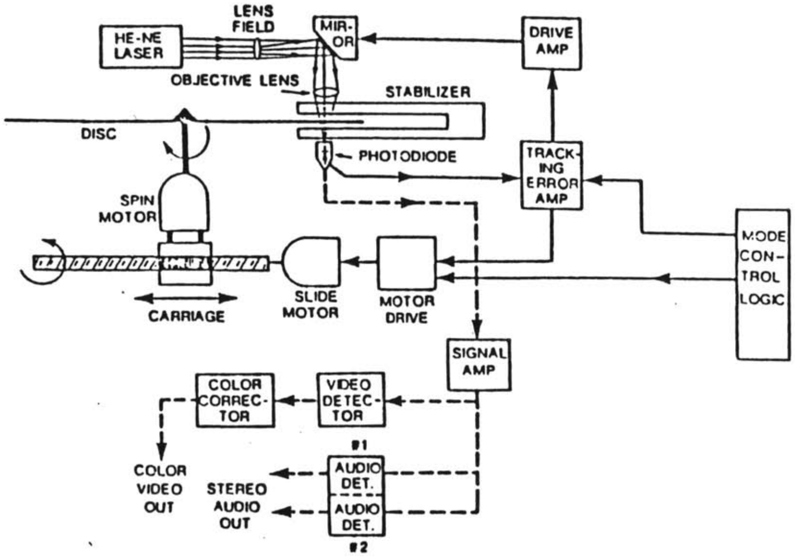
However the quality of design cannot be measured in these terms and the validity of particular solution depends primarily on qualitative issues or issues which are quantifiable but where the rules for this quantification are particularly complex and dependent upon many other factors. In a few words heuristic knowledge is important, in order to come up with viable suggestions for these complex problems.

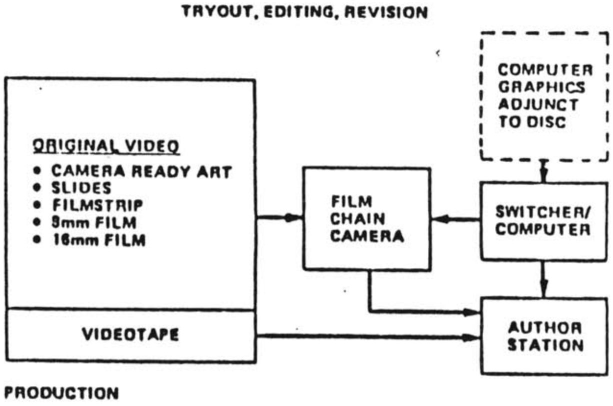
Most research in issues related to computer aided design and any issue related to the way a computer system can assist the architect



Video Disk : 3 Choices Source : Videodisks: Three Choices IEEE Spectrum :38-42, March 1982

VIDEO DISC PLAYER BLOCK DIAGRAM





1. Compile Slides, Artwork, Filmstrips for transfer to master filmstrip.
2. Add motion picture film and/or videotapes, if desired, for transfer to master 1" or 2" videotape.
3. Input for Videodisc master.

Top : **Video Disk Player Block Diagram.** Source : Frontiers in Education Conference. Miller 1978 **Bottom :** **Production Sequences.** Source : Therese Vien, September 1985 in decision making towards a particular design solution, or at least evaluate a given solution, is therefore focusing in the area of expert systems and knowledge engineering.

Expert systems have been previously described. They can be used to assist in analytical or synthetic work, however the latter tends to prove more difficult. They all have a very narrow area of expertise, which is based on a large knowledge base containing information and rules for how this information is to be linked together and utilized. They can be structured in different ways, as rule based systems, frame based systems etc. However their function is always the same, i.e. to be able to assist the user by providing expert advice within their limited area of expertise, to be user-friendly in their interaction with the user, to be able to justify their answers, to be able to learn from experts and in short to act the same way as a skilled human consultant would.

The most serious limitation of expert systems today that prevents them from being particularly useful in practice is their very primitive mechanisms for input of data in the process of building the knowledge base. It is becoming more and more conscious that the lengthy and tedious brainstorming of experts in the field and computer scientists is a serious handicap for the implementation of any expert system.

In the case of architectural design, there are many more reasons that make this scenario difficult to implement. Design is too complicated a task to be able to decompose it objectively, decide on the rules and input them consistently in a knowledge base.

In the context of the open system for architectural design the expertise is too stored in a modular way and distributed all over the system. There are no large elaborate systems that deal with the immensely complicated issues related to architectural design. Instead there are several small expert systems that are built to decompose the most simple puzzles that an architect encounters during the design process.

Each such "mini" expert system is called from the drawing editor in order to evaluate, advice or interactively help adjust a particular issue for which it is designed to provide advice. Access to these facilities should be very easy and provided directly from the graphics editor.

These facilities are created within a given "shell" for the design of expert systems. Since even the smallest expert system in order to be efficient and of use in professional applications must contain a relatively large amount of data, it is likely that both the design of such systems, as well as their use, will have to rely for the immediate future on the use of minicomputer facilities.

Ideally, in the long run, as hardware inference engines for the design of the system and input facilities improve, such issues will be resolved and the key consideration will be fast access and modularity of the various components. As more of these modest expert systems are built to assist designers in the most trivial routines within the design process, expertise will be building up and more ambitious expert systems might be built based on this expertise, the mistakes of the past and the extensive "libraries of existing small systems" that have been already developed. Also users by that time will be familiar with the use of such tools and the extent to which they should consider their advice.

In the case of the development of routines which can be thoroughly improved by the incorporation of a considerably limited amount of expertise within a set of instructions, these routines can most likely be developed within the programming environment of the graphics editor. This very primitive model of storing expertise within a computer program should not be overlooked since it is particularly useful in the cases where rules are simple and straightforward. Access time is much faster, flexibility of the tool much greater, and for the near future hardware required for these applications much more affordable. If the programming environment within the graphics editor is some version of lisp or some other language appropriate for artificial intelligence applications, this task is likely to become much easier.

4. SCENARIO FOR THE USE OF THE COMPUTER BASED BUILDING SYSTEM

4.1 Introduction

This chapter attempts to briefly describe the essential features of a C.B.B.S. The use of the C.B.B.S. presupposes the existence of an Open Computer System for Architectural Practice which has been described in the previous chapter.

The system assists the user to rapidly become familiar with the rules of the building system and to design the project with the minimum waste of resources. The ultimate goal is to increase the efficiency of the design process by ensuring consistency, and eliminating time and cost of the process.

The Computer Based Building System initially supports the user in the selection of the appropriate subsystems among a list of options, whenever such options exist. It is also responsible for helping the user become familiar with the essential rules of the system in an interactive tutorial. These tasks are referred to as preliminary tasks since they are performed before the actual design of the system is initiated. The goal is that these tasks are suppressed as much as possible to the benefit of the nest stage which involves the actual design of the project.

The next stage commences with the schematic design of the project. This stage is critical since several alternative design schemes are considered and evaluated. The goal is to force decisions towards the initial stages of the design process. This implies the existence of a separate set of evaluation routines which are appropriate for evaluation at the schematic design stage with only a limited amount or relevant information.

The schematic design stage is followed by the detailed design stage, which is automated to a large extent, depending primarily upon the nature of the building system utilized. This stage is followed by the final stage which is dedicated to the production of the construction documentation. This step is totally automated.

All stages of the process depend on interactive graphics, which provide the link with the Open Computer System for Architectural Practice. The use of the system is based upon an interactive session with the computer where the rules of the system become apparent to the user without diverting him from the design for long tutorials on rules of a system which he is likely to never again utilize.

The goal is that the user does not have to make a major investment in time in order to utilize a specific building system. Instead, the utilization of a building system in a particular project implies considerable savings in design time. The project architect is more willing to utilize an "open system" approach. since the incorporation of each system in the building does not require specially trained personnel. Thus the use of industrially produced building systems can be generalized and the size of project and firm shall no longer restrict their use.

4.2 Preliminary tasks

4.2.1 Selection of compatible subsystems

This feature is optional and is only relevant in the case of Computer Based Building Systems which contain a large number of subsystems, as in the case of a large closed building system. In such cases the user is offered an option among several alternative subsystems which perform a specific function, e.g. partitioning or plumbing, according to the specific needs of the project he is required to design.

Clearly the selection of these subsystems is the first step in the decision making process the user has to go through, before he can further utilize the C.B.B.S. and proceed with the design of the project. All subsystems are compatible since they are part of a closed system, but the user has to be informed on the individual features of each system in order to make a rational choice.

The system provides the user with relevant information on each of these systems, and helps him decide on which option suits best his individual needs. This facility in its most primitive version can simply consist of a non interactive tutorial containing slides and textual information, hopefully with the assistance of the visual information system, or at best can be some kind of expert system which will interact with the user and justify its suggestions.

4.2.2 Interactive Tutorial

Once the set of building subsystems to be utilized in the project have been selected, the system is responsible for providing the user with more information about each subsystem to be utilized. This is achieved by an interactive tutorial in which the user can select the areas of information that are of particular interest to him.

However, the flow of information to the user at this point must not exceed a certain amount. The goal is not to replace the manual of the "paper based" building system with a electronic manual. Instead the user is provided with only the information that is necessary in order to become familiar with the basic rules of the system to the extent that these will influence the design concepts to be followed.

The Computer Based Building System aspires to provide the user with information implicitly during the design process by appropriate grid and snap facilities, instead of explicitly teaching the project architect the rules of the system. Therefore the interactive tutorial only aims to cover the information that the designer needs to be aware of before the design commences.

The tutorial is based on appropriate combination of textual and visual information. The system ideally utilizes both slides of vector drawings, and the visual information system described on the previous chapter which is based on video technology, however can be utilized alternatively even without the need for the extra expense which required for the hardware of the V.I.S. The system can display either single slides which contain information that the user can read and provide some feedback, or can display video sequences which demonstrate construction sequences, exemplars of buildings etc.

4.3 Schematic Design

4.3.1 Introduction

A sketch facility enables the user to construct rapidly a three dimensional representation of the building. The only information that is represented at this stage is related to the volume of the building and even that is not yet accurate but indicates the users intentions in very general terms. The volumes can either be polygons, extruded in some height in the third dimension, or in more sophisticated systems and according to the needs of the particular building system are likely to be polyherda which implies a true 3D facility.

When nature and prospective application of building system allow for standardization of functional units within the building at a scale that affects the volume and the layout of the final building, libraries of such standard volumes are also available to the project architect.

The system produces a wire frame drawing that can be seen in isometric and perspective modes from any defined view-point. Hidden line removal facility useful but not essential.

The system also provides for a sequence of routines for brief calculation of construction costs for each alternative proposal based on the building subsystems selected, the volume of the building etc. Also a set of applications that perform very brief thermal calculations and eventually provide the architect with a brief estimate of the maintenance and running costs for each alternative solution.

The user has therefore the opportunity to rapidly construct, view, review and evaluate several alternative design schemes for the specific project in a very short period of time.

4.3.2 Creating the alternative schemes

Since the user has already decided on the preferred structural system to be used or other subsystems that are critical for the modular coordination and dimensioning of the building, a variety of tools are available in order to assist the user at this stage to come up with a sketch design that is realistic and easily implementable by the selected building system. Grid and snap facilities for example might be utilized to indicate optional floor heights, widths of wings of the buildings etc.

It must however be stated that these indications should not be restrictive since the user might wish to include non systemic parts of the building in the schematic layout design.

The above mentioned facilities are meant to constantly remind the user of the rules of the building system chosen. However there are several instances where the user might be willing to be further guided and restricted with functional constraints irrelevant with the structural system. This is normally the case when the building system has been specifically designed for a particular type of building e.g. school, hospital, etc., as was the case with several computer aided building systems developed in the past. In this case it is also likely that a library of standard volumes representing functional units of some scale will complement the sketch design facility allowing the user to insert these already defined entities instead of building his own volumes. These volumes may represent classrooms, hospital departments of various kinds etc. acording to the function of the building.

These standard volumes though simple in their representation will correspond to well defined to the smallest detail building descriptions, already stored in the system and presented to the architect upon request in a later stage of the design process.

This approach of treating complex functional units of large building complexes as "black boxes" is very efficient and particularly suitable in the context of a computer based building system. The principle here again is that the project architect has no need to reinvent the wheel. The systems design team that has come up with a list of specifications and several optimal design layouts for a specific hospital department given a particular building system is probably more suitably equipped to make decisions based on a large data bank of relative information than the project architect who has only general knowledge of the subject and cannot be involved with all the relevant technical details. The role of the project architect is instead to link these functional units together along a circulation spine in a functional economical and aesthetically pleasing way which is enough of a problem in itself.

4.3.3 Evaluation routines

At this point a sequence of sketch design costing subroutines are initiated with the task of producing very rough estimates of the cost of each alternative solution according to a set of parameters associated with the building subsystems chosen, the results of the site analysis, the volume of the building the total exposed area of the building etc. These estimates include the cost of materials involved in construction, construction costs, and maintenance cost of the building, a fact which calls for several routines that perform thermal analyssis etc.

It is obvious that it is much easier to reach very accurate results from these estimates if the system is dedicated to a specific building type and the layout is an assembly of predefined functional units which need a well defined set of elements for their assembly and have a known construction and maintenance cost. (Harness system.) However if a system is not dedicated to a specific type of building it is probable that it will still incorporate a library of predefined prototypes of spaces for various functions that could form a basic vocabulary for the project architect to utilize. If the largest part of the layout consists of such library parts the estimate is likely to be very accurate indeed. It must be noted that this library of functional units can be updated and expanded with every finished project that is built by the system.

4.4 Detailed Design

4.4.1 Introduction

The amount of design decisions still left open for discussion at this stage depends upon the nature of the particular building system as well as the approach that the project architect followed during the sketch design.

If the specific project and/or his creative impulses didn't leave much room for restriction within the limited vocabulary of functional types provided by the libraries of the system there is still a lot of work for the designer at this stage. The system then guides the architect during this process ensuring that systemic rules are not violated, and provides the designer with any information he may require.

If on the other hand the project architect used to a large extent functional units with known and already defined contents then the design work is already essentially completed. The designer only has to concern himself with the non systemic parts of the layout as well as any variations or beautifications of the original solution he may find appropriate.

4.4.2 Production of the detailed design of the project

In this stage the user of the CBBS needs primarily two tools that will assist him in the production of a final design that is efficient, without excessive waste of resources. Both tools ideally operate in the same manner, hopefully within the same graphics editor with the operation of which the user is already familiar but are suitable for performing radically different tasks.

One of these tools is ideally designed to assist the user in designing the systemic parts of the building, allowing for several levels of automation of the process according to the scale of the building, its use, particular conditions of the site as well as the extent to which the project architect wishes to automate the process by utilizing previously defined functional units and avoiding deviations from the rules of the system.

The second tool is suitable for the design of the non-systemic parts of the building that are likely to exist for reasons of efficiency and appropriateness of available alternative systems within the particular local market. These "non-systemic" parts of the building which are likely to be proportionally increased with increased total area of the building layout as well as complexity of the building program, exist up to this stage as defined "black-boxes" within the overall building layout.

This tool will provide the architect with infinite flexibility in order to design them efficiently. It will appear to some extent like a conventional graphics editor, however it has to be both suitable for architectural applications, i.e. appropriate for the insertion of walls, doors etc., while at the same time very powerful and not restricted to orthogonal representations. It should allow with the same ease both the design of an orthogonal room with standard walls and of a room enclosed by several walls of varying thicknesses placed at random angles.

This facility is in most cases likely to be part of the Open Computer System for Architectural Practice, permanently resident within the firm and produced independently from the C.B.B.S. However, the C.B.B.S. should contain some module, however primitive, that performs these tasks, or should be designed in such a way to easily interface with a similat tool produced by some other independent manufacturer.

Besides the help on line through the terminal and the explicit indication of the rules of the system at every stage of this process, several other interfaces have to be developed in order to provide the user with additional services at this stage. These include access to spreadsheet for calculations, as well as other analyses and evaluation packages. They also include access to the apprropriate information about the building system, the project site and the legal and other constraints with which this site is loaded. More advanced facilities at this stage would include access to expert systems which would provide expertise for particular design tasks. This implies the existence of a kind of a Decision Support System for the designer.

An other useful potential interface at this stage is that with the Visual Information System. The Video Disk serves as an archive for storage and retrieval of large quantities of visual information which can be displayed in several modes upon request at any stage of the design process.

This last application seems to be very useful since it can be utilized to display slides as well as video sequences of systemic components, construction sequences as well as buildings constructed by the system, details of the buildings, finishes etc. It can also be utilized to retrieve images unrelated to the building system itself such as exemplars of buildings of various functional types, of buildings of a particular local context etc. The user of the system will ideally have access to a large database of such visual information which will assist him in his design.

However it is important to have in mind at this point that the user should have access to these facilities without the need for any capital investment in either hardware or software besides the equipment he is already familiar with, and has already access to, in his current practice. It is hard to imagine a project architect happily deciding to utilize, and more important efficiently using, a computer system that is totally new to him, for the design of a single project of a relatively small scale, unless some components of the system are already familiar to him. As appropriate such components for an architectural firm, one immediately identifies the graphics editor as well as some of the application packages which can potentially be parts of the O.C.S.A.P. It might well be that components of specific hardware are provided together with the complete software package (i.e. the C.B.B.S.) by the building system supplier to the project architect that is committed to designing a project with the use of the specific building system. And indeed, one may argue that the supplier has considerable incentive in doing so especially if the project is relatively large. Some of the software might be provided in house, while some other, like extensive data-bases could be accessed through the network or at worst through a modem.

However in times of peak in the demand for the system, as well as in instances when the project to be designed is relatively small and therefore the task of designing relatively trivial, it might not be possible or even appropriate for the systems manufacturer to supply the whole package. The system should be built around the graphics editor in a modular way, so that several components - modules - can be added for the benefit of the overall package and the facilitation of the task of the designer. However these components are not be mandatory for the use of the system which is flexible with a wide range between a "minimum" and a "maximum" package.

4.4.3 Evaluation routines

Evaluation at this stage consists of a package of routines accessible from the graphics editor that inform the architect - user of the CBBS of the performance of his design. Such routines involve costing, daylighting and artificial lighting analysses, thermal analysis, which gives the user some indication of the maintenance costs of the building, structural analysis etc.

In the future and as research in the area of knowledge engineering advances one will hope to see some expert systems which will be able to read graphics information and be able to produce some evaluation of the actual design of the project or at least be able to answer specific questions.

At present however, the generic differences between this set of analysses and the previous one in the schematic design stage is that now there is much more data to be analysed and therefore a much higher expectation of accuracy from the results and that these results will be utilized for the evaluation and further improvement of a given solution until a standard that the architect considers satisfactory is reached and not for the evaluation of alternative solutions. It is obvious that these new programs are much larger and take more time to run than those utilized in the previous stage.

4.5 Production of Construction Documentation

4.5.1 Introduction

The production of construction documentation is the part of the process of using the CBBS when the system takes over almost entirely and relieves the user of a large amount of repetitive tasks that he conventionally had to invest time and effort on.

This stage involves the production of the construction drawings, the production of bills of quantities of the components that are utilized in the proposed building, a task particularly easy in the case of component building, the production of the appropriate documents for the ordering of these components and materials, the production of a construction schedule as well as a set of construction specifications.

4.5.2 Production of final construction drawings

This process is ideally totally automated. Once the detailed design phase is completed the user has simply to answer a few questions regarding the size and format of the output as well as the contents of each required drawing, and the drawings are produced automatically. The elegance with which this will be achieved has more to do with the sophistication of the drawing editor that serves as a base for the CBBS than the specific overlay specifically designed and provided by the building system supplier.

The system, through layering techniques has the potential of creating drawings of each subsystem separately and at any scale the user specifies. The user can also select the media and the appearence of the output from a wide range of alternatives. It has also facilities that automate to a large extent laborious repetitive tasks, i.e. dimensioning, etc.

In the case where present limitations of the micro-environment prevent the system to be properly set up on a true 3D system that builds a model of the building, then the production of the actual construction documents might prove a little more laborious prompting the user a little more at the beginning and requiring a separate interaction between the user and the terminal for each drawing to be created. Still this limitation, which with the current rate of developments in the area can safely be anticipated to no longer be a problem within the next couple of years, will allow us to have a production of construction drawings almost entirely automated.

4.5.3 Production of reports: Bills of quantities - Specifications

The various analyses, invoked as a last stage of the detailed design phase have produced already a sequence of well formatted reports that give a good overall description of the performance of the building as finally designed.

Of these, one must particularly stress the bill of quantities that is produced automatically by the system and which leads directly in a total cost of construction for the systemic parts of the building. The system will ideally also calculate approximate cost for the non systemic parts of the building according to latest estimates on material costs. It will produce well - formatted results from within a spreadsheet environment. Furthermore if the user desires within a spreadsheet environment it is possible to obtain more figures related to the suitability of the investment, the maintenance costs, the internal rate of return etc.

The system will also generate a set of specifications. These, should ideally be performance specifications for any non systemic part of the building or any sub-system that is not explicitly defined within an already selected set of closed building sub-systems which are an integral part of the CBBS. The system will automatically produce any documentation that is required to describe the systemic parts of the building and will assist the user through a specification editor and a data-base of specification clauses on the fact and efficient creation of a appropriate set of performance specifications.

5. COMPUTER FACILITIES WITHIN AN O.C.S.A.P. WHICH COMPLEMENT THE COMPUTER BASED BUILDING SYSTEM

5.1 Recording and maintaining project information

5.1.1 Recording project requirements and constraints

The first thing a project architect is urged to do once he is faced with a potential job, is to collect and systematically store project requirements and constraints in order to first evaluate his potential interest for the specific job, as well as utilize this information for the design of the project.

The information at this stage might include programmatic requirements, desired function of the building, brief information on budgeting and time constraints, indication of the site, as well as other specific requirements of the client. This information at this point is likely to be available in printed format of some kind and may even include some drawings.

5.1.2 Recording user needs

In a similar way the project architect utilizes a data- base facility to record and retrieve upon demand all information that is relevant to the needs of the user, his intentions etc. This facility helps him both in the design of a project that is more suitable to the users need, but also ensures that no waste of resources will occur due to misunderstanding of these needs.

From an early stage the architect records every meeting with the client on the system, and later has the chance to observe potential inconsistencies, and bring forward the issues before it is too late and resources have been wasted in work that was not done in the direction the user suggested.

This facility also is used to protect the user against impossible clients from an early stage, either by helping the architect realize the incompatibility in the clients demands, or by providing evidence that he indeed has been working in the direction mutually agreed.

5.1.3 Recording Site Information

This facility allows the architect to record consistently and systematically information related to the site of the project. This information can be updated continually, as new evidence appears, and in a multi-user system can be available continuously to all members of the design team.

The most important feature of this facility is that it allows all members of the design team, whether they have visited the site or not, to be equally informed. The project administrator decides on the relevant information that should be on line for the members of the team, and the system provides a common base for communication for the team.

5.2 Building System Selection

5.2.1 Alternative Scenaria

The first step in the decision making process that will eventually lead to an efficiently designed product and consequently to a successful building in terms of user satisfaction and minimization of costs for both construction and maintenance has to do with the selection of the appropriate building system to be utilized.

Several factors have to be taken into consideration in this process, including issues of available raw materials and technology, cost and particular skill of labor in the area, as well as distance of the construction site from production areas, quality of transportation network and time available for construction. It also involves issues related to local building regulations, climate, and cultural preferences of the prospective users.

These issues are often hard to deal with effectively since they are to a large extent interrelated and involve a complicated optimization process as well as decision making with limited information. The answer to the problem depends primarily on facts related to the site which cannot be dealt with efficiently by using a generic facility. Although the issue is not easy to tackle, this initial decision on the building subsystems to be utilized is probably the most essential step that influences the future performance of the building. To a large extent, most of the following decisions are based on it.

Conceptually such questions can only be answered through a very interactive process with the computer utilizing the current developments in the field of artificial intelligence. This can only be achieved by an expert system that interactively guides the user through a selection process and help him or her eliminate options in an efficient way.

This expert system can only be developed by some central organization which has interest in maintaining a high standard in information communication. Once the core of the expert system containing the rules is developed it can be potentially linked to a number of separate knowledge-bases each one containing information related to a specific local context. The project architect utilizes the expert system linked with the appropriate knowledge base or when in in doubt about the suitability of each of these knowledge bases repeats the process linked to different knowledge bases of neighbouring localities and compares results. It is assumed that these data bases are all similarly structured following some national of regional standards and will contain information related to local building regulations, cost of raw materials and labor for each trade, distances between localities and production sites of industrial systemic components. etc.

The most difficult task of a project architect today is to be able to keep track of all the available products in the market and their specifications. This task often tends to become increasingly difficult and therefore decisions are made based on the limited existing information about a few products, excluding better options that were not known to the architect, often with disasterous effects. In the case of systems building the architect has only got to make decisions about optional systems and subsystems of components, a much easier task.

The architect may alternatively choose to utilize a relatively closed building system in which case he has already limited his choice of subsystems to those provided by the building system, or prefer to utilize a more open scenario, in which case he is likely to select compatible subsystems from the open market. Whatever the case he is likely to be guided and helped considerably by a C.B.B.S.

5.2.2. Restrictions and present day limitations

The development of a sophisticated Decision Support Computer System which assists the project architect in the selection of building sysetms and subsystems from the open market is a very important contribution that will greatly assist the application of Systems Building in Architecture.

It is however understandable, that this is an undertaking of considerable scale. The problem is not so much in the development of the expert system itself but in developing and maintaining the various independent data-bases that are to be linked with the expert system. One may say that despite its great potential utility such a system is not likely to be created unless a there is organized action from various bodies for this purpose and it is not wise to assume that the existence of such a system would be a given at any context, however developed, in the near future. This is because it requires a considerable waste of resources at a regional level for the creation and maintenance of the various data-bases as well as very good organization at the central level for the development of the rules of the expert system as well as a process ensuring consistency among the various data bases that are going to be developed at a regional level.

5.3 Designing the Non systemic Parts of a Building Layout

When a set of compatible subsystems are selected from the open market or as part of a closed building system, to be utilized in a particular project, this does not necessarily imply that they are going to be utilized throughout the project. Some portion of the project might well be excluded for functional aesthetic, or other reasons related to the morphology of the site etc. This forms the non systemic part of the building which can be concentrated in a specific area of spread out throughout the layout. Non systemic parts of a building can be for example an elaborate staircase in the middle of the atrium of an office block, or the ground floor of a hotel building containing the reception, or all the staircases of a particular housing project.

The non systemic parts of a building layout are identified early in the schematic design process. However, in order for the project architect to be able to efficiently design these parts of the layout and to produce rapidly and effortlessly the appropriate construction documentation, a different environment is required from that for the rest of the building.

The overlay of the graphics editor that is suitable for the design and production of the construction documentation for the systemic parts of the building layout which is provided as part of the CBIS of the CBBS, is not useful in this respect. An other generic overlay has to be utilized instead, permanently resident within the architectural firm as part of the Open Computer System for Architectural Design.

This overlay to the graphics editor would allow for easy design in an environment where the user does not accept restrictions from the system related to materials dimensions of components and modular grids. The user does not assemble the building form a library of parts. Instead he is allowed to freely input walls and elements of any size at any angle and has a menu of powerful commands that enable him to develop and view his design rapidly and efficiently.

SECTION 3. IMPLEMENTATION :

Development of Components of the C.B.B.S and the General Computing Environment for Architectural Practice

SECTION 3. IMPLEMENTATION : Development of Components of the C.B.B.S. and the General Computing Environment for Architectural Practice

CONTENTS :

SECTION 3. IMPLEMENTATION : Development of components of the C.B.B.S. and the General Computing Environment for Architectural Practice

1. INTRODUCTION

This section includes the description of several computer programs written over the past 18 months, which clarify some of the principles and techniques already described in this document.

It is hoped that through the description of these programs, the reader will be able to acquire a better picture of the use of a Computer Based Building System within an Open Computer System for Architectural Practice as well as how some of the components of these systems can potentially operate.

Furthermore the description of these programs which will follow, is structured in such a way so as to give the reader some idea of the techniques utilized in the design of these facilities as well as the various design considerations which underlie their design.

It must be made clear however that these programs do not form a fully blown computing environment for the application of systems building in architecture. On the contrary their development has been limited by the existing facilities in the Computer resource laboratory of the school of architecture and planning, as well as the particular interests of the author, and was dictated by the desire to clarify those principles that were critical for the comprehension of the essential features of the whole environment, i.e. the modularity of the various components of the system, the appropriate division of responsibilities etc.

These programs are in some cases presented as a sequence since the process for their development was limited in each case by the available programming facilities within the environment of the modular graphics editor that was utilized.

This mode of presentation was selected intentionally since the same process will characterize the actual full blown implementation of these concepts, where several hierarchical levels of programming will exist with different responsibilities assigned to the programmer at each level and the programmer at the lower level will be constrained by the programming environment set up for him by the top-level programmer.

As already described in the previous section it is this differentiation of distinct "levels of action" which will enable the user of the system i.e. the project architect, to satisfy his varying needs for computer support in the complex process of design.

Two of the programs whose description will follow are dedicated to component building and through their description both the reasoning that lead to the adoption of the parallel development of the components of the building system with the computer based information system, as well as the design considerations for the development of the graphics editor resident part of a C.B.B.S. will come through.

The rest of the work can be conceived as supplemental components of a C.B.B.S. forming part of the Open Computer System for Architectural Practice which will be permanently resident within an architectural firm.

This program is a customization of the modular graphics editor to make it appropriate for the design of projects or parts of projects which due to their particularities, practically do not allow for constraints originating from functional standards or standardized materials and components to be imposed to the designer. This feature makes this facility particularly useful for the design of the non systemic parts of a given project.

Since these programs were developed within the environment of a modular graphics editor which provides an internal LISP programming capability and they are meant to operate together with the graphics editor in order to enhance it and make it suitable for a specific application, they are in a sense developed to be overlayed to the graphics editor which shall serve as some appropriate operating system for the use of these programs. Therefore these LISP programs are referred to extensively as overlays to the graphics editor in the text that will follow.

2. SYSTEMS BUILDING - MODULE 1

2.1 Introduction

2.1.1 General Information

This work is an overlay on AUTOCAD'S graphics editor that makes it a suitable tool for the production of 2D representations of a building which are consistent with the rules of a given closed building system.

It incorporates a structured menu which allows for the selection of subsystems within the closed system. The package offers help on screen which indicated explicitly to the user the rules of the system, as well as features which allow the user to be aware of the rules of the system when designing without explicitly having to inquire about them, e.g. appropriate snap and grid facilities etc. It also offers an attribute extraction capability to a spreadsheet environment for calculations and manipulation of alphanumeric data incorporated in the drawing.

This facility was developed as an overlay on AUTOCAD versions 2.00 and 2.01, during spring 1985. These versions of autocad did not incorporate any capability for 3D representation of a building and did not offer a true programming environment within the graphics editor.

The programming consisted of the development of an appropriate menu by utilizing a set of conventions understood by AUTOCAD, and the development of script files, i.e. text files that invoke the execution of sequences of autocad commands. The facility also required the development of the appropriate drawing files and slide files which support the menu. The programming environment was indeed very poor, practically non existent, but appropriate utilization of various system variables set by the graphics editor in combination with sophisticated use of commands in appropriate sequences enabled enabled remarkable results.

However this project was an early demonstration of the need for programming within a graphics editor in order to produce a graphics editor that is flexible and suitable for individual needs and varied applications. The experiment proved that even within a poor programming environment the customization can produce impressive results that increase the potential of the existing graphics editor for a specific application and particularly for systems building.

The most important outcome however from the development of this facility came from understanding of the limitations of attempting to create a computer based information system that supported an already existing structural system. The rules of the existing system were very difficult to implement in the computer system without straining the user and significantly delaying the design process with escessive zooming in and out of the drawing for the insertion of various components. The modular coordination imposed by the building system designer was so restrictive and the rules so complicated that they imposed a considerable strain on the whole facility in terms of the sizes of the drawings that needed to be supported, the size and number of the programs that had to be developed but above all the complication of the process of interacting with the user that resulted. The result was a compromize among the variable spans of the system which were retained in order not to reduce its flexibility, but a modification of the dimensions of the structural components in order to facilitate the interaction with the user.

Thus, very early during the course of this exercise the limitations of this approach caused the deviation of the initial goal and a move towards the parallel development of a computer system and a building system which was more fully realized in the second module - overlay on the graphics editor for systems building, developed in the fall of 1985. The advantages of this second approach have been extensively documented in the preceding sections of this document.

2.1.2 User requirements

In component building, a building is assembled from a number of systemic elements that belong to various compatible sub- systems. Each building System (or subsystem) implies the existence of an array of such systemic elements along with a set of rules for their assembly. The dimensions and properties of these components, the rules for their assembly s well as interfaces with elements of other subsystems, have been thoroughly examined and tested before industrial production of the system commenced.

The task of the project architect is not to redesign, alter or modify these elements in any way, but instead to make himself familiar with the rules of the system and to design as rapidly and efficiently as possible the desired layout that best fulfills the building program without violating any of the systemic rules.

Thus, in a systems building environment, unlimited "freedom" is no longer considered an asset but instead it is both dangerous and inefficient. Dangerous since it is likely to lead to a deliberate or accidental violation of the systems rules, often with disastrous effects in the interfaces and assembly of the building components, and inefficient, since it slows down considerably the work of the project architect.

The primary role of the overlay to the graphics editor is to enable the user of the computer system to become familiar with the rules of the building system without the need of a reference manual.This can be achieved both directly through the use of a sophisticated help on screen system, and indirectly through a number of facilities like appropriate grids, as well as snap and layering facilities that would indicate these rules and prevent the user from violating them.

Thus, within a computational environment appropriate for systems building the user must be able to perform the following tasks with reasonable ease:

a. Select compatible building subsystems among a list of options.

b. Become familiar with the rules of the system. (Help on screen etc.)

c. Design interactively on the screen during sketch design stage.

d. Produce rapidly, accurate detailed construction drawings directly after the sketch design phase without the need for tedious insertions with excessive zooming.

e. Test flexibility of solution as well as compatibility with the building program.

f. Analyse results for each potential solution, compare cost etc. in order to reach objectively an optimum decision.

More specifically when working on a micro-computer environment it is important to also take into consideration, issues related to memory consumption and storage space available. This often turns out to be an important issue since drawings tend to occupy a lot of space for their storage. However, it is generally easy to develop strategies in order to overcome such problems when working with a more or less closed building system and a finite number of elements.

2.2 Significance of the project - Design Principles and Techniques

2.2.1 Significance in the Conceptual level

The development of this facility aided considerably the acquisition of several insights on the potential of a modular graphics editor that is customizable the user requirements from the internal programming capability, the usefulness of this environment to systems building as well as the importance of parallel development of a physical building system and a computer based information system that contains the rules for the system.

More specifically the significance of this facility in the development of the conceptual basis for this work can be summarized as follows :

- It showed that a poor programming environment and inappropriateness of already developed B.S. results in an increase in the size and the number of the programs and the drawing files that constitute this facility.
- It offered possibility to evaluate importance of good programming environment as well as what features of such an environment are more urgently needed.
- It offered possibility to anticipate the need to develop building system and computer system in a parallel process in which case both parts are likely to be more efficient.
- It indicated a scenario for the development and the structure of a menu that is appropriate for a closed building system of considerable size containing several compatible subsystems. It also indicated the potential magnitude of such a facility and the need to develop strategies to economize in memory consumption and storagc space.

2.2.2 Significance in the Practical level

This facility is an overlay on AUTOCAD'S graphics editor that makes it a very efficient tool for the project architect or the student that would like to use such a closed Building system to design a building. It is particularly efficient in the rapid production of 2d drawings and its power is limited by the programming facility available at the time within the graphics editor as well as other generic limitations of this editor (2D).

More specifically it allows the user to perform the following tasks:

- To use the higher levels of the systems menu in order to select a set of compatible subsystems to be utilized in a specific project and test their flexibility and cost effectiveness, or to acquire through the screen information on the rules of these systems.
- To use the main part of the systems menu, in order to design relatively small floor plans, insert assemblies of elements and components of various subsystems (Structural, partitioning, cladding, plumbing, mechanical, HV-AC etc.)
- To use the automatic insertion option in order to quickly assemble the structural system of large buildings, through a very swift process that leads directly from the sketch design stage to the production of accurate construction documents.

2.2.3 Design principles and techniques indicated

During the course of this work the necessity for the utilization of several techniques for the design of the system became apparent and the facility was designed accordingly. The most important of the resulting design principles will be mentioned here.

A different approach is required for the user to be able to efficiently use the system in projects of different scale. Several building systems or subsystems are flexible enough to allow them to be efficiently utilized in projects of very different magnitude. However the C.B.I.S. has to adapt in order to meet user requirements which change considerably according to the type and most important the size of the project to be designed.

In the project this approach is demonstrated in the development of the structural system. The user according to the scale he is in can either utilize to insert the structural components by selecting them form the library of parts or by picking assemblies of components and inserting them through scripts which facilitate his work, or, can alternatively utilize the automatic insertion capability which inserts whole wings of buildings automatically and is appropriate for much larger building layouts.

An other important consideration already mentioned in the user requirements is that of reducing memory consumption and storage space particularly when working within a microcomputer environment.

In this facility large savings were achieved by the technique of utilizing scripts to create complicated drawings from simple parts and insert the assemblies upon request of the user. Thus with only a few simple parts in the library the user is able to insert with a single command a complex drawing which would have required a large amount of space to be stored. Instead a script file was saved which required a very small fraction of the space required for the drawing.

The scripts were designed in such a a way as to be easily updated with a new version of autocad, a fact which was proved in practice by the fact that this overlay was up and running with autocad 2.15 during the summer of 1985 with less than a day's work. The savings in storage space were considerable since the number of the complex drawings that would alternatively have to have been created form the simple blocks would have been many more than the simple ones, and these savings allowed the facility to be stored on a single high density floppy disk.

The introduction of the optical disk for storage of information will obviously provide a more appropriate and permanent solution to memory considerations, particularly for systems building applications which involve data that does not require to be changed or updated and is part of the master file.

One other important area that has been investigated is related to appropriately structuring and nesting the menus that appear on the screen and form part of the systems menu in such a way so as to be both helpful for the user and compatible with rules set by autocad.

Here we shall be particularly concerned with techniques that are of help to the user since these are of more general interest regardless of the limitations imposed by the environment of the specific graphics editor.

The hierarchical structure of the screen menu is particularly helpful for systems building applications. It is particularly useful for helping the user become familiar with a particular system since each "page" contains only a few commands and this allows the user to anticipate easily his next move by selecting one of the few options available to him which appear on the screen. On the contrary the tablet menu is better for acquiring an overview of the multitude of commands but by its structure does not impose any hierarchy to the user and therefore does not guide him efficiently in which of these options he is allowed to select at any given moment.

Therefore since already in section 2 of this document it was established that interface to a specific building system is very likely to be utilized only once by a particular user and therefore it is not expected or even desired for him to acquire experience in using the system, the screen menu provides the best solution. Furthermore it can practically store much more information than the tablet menu.

In the context of this facility for systems building the tablet menu is of particular utility only when the user is prompted to select from a large library of components of a given subsystem. However ideally in the design of the system this will not occur and here again is one example where inefficiency of the physical building system - too many components ans therefore increased production cost - is matched with inefficiency in the C.B.I.S. which requires the utilization of a second medium which will attract the attention of the user besides the screen. In both modules for systems building developed in the context of this work, this did not occur, and the screen menu was sufficient to convey the required information. However it is likely that in large and sophisticated production systems the tablet menu will also be utilized as a master template containing all the components grouped in clearly defined areas according to the subsystem they belong in.

Once the utilization of the screen menu instead of the tablet menu was decided several issues had to be resolved related to its potential structure. The results of this investigation can be summarized as follows:

- Excessive nesting of menus is not desirable however it is often difficult to avoid.

- The structure of the menu has to depend exclusively to a well studied, detailed scenario for the potential use of the facility and the anticipated moves the user will wish to perform more frequently.

- Making the primary assumption that the user will spend some time working on one subsystem before moving on to the next one, the approach advocated is to facilitate the work within each subsystem as much as possible.

- It is important to establish a "primary" menu for work within each subsystem with which the user will soon become familiar and from there let him explore the various optional facilities offered at different levels of the screen menu.

- If the level of nesting is considerable within the menu these "primary menus (one for each subsystem) should ideally be located approximately half way down the hierarchical structure of the menu in order to facilitate access to other "satellite" menus.

2.3 Detailed description of the facility

2.3.1 Design of the facility - Structure

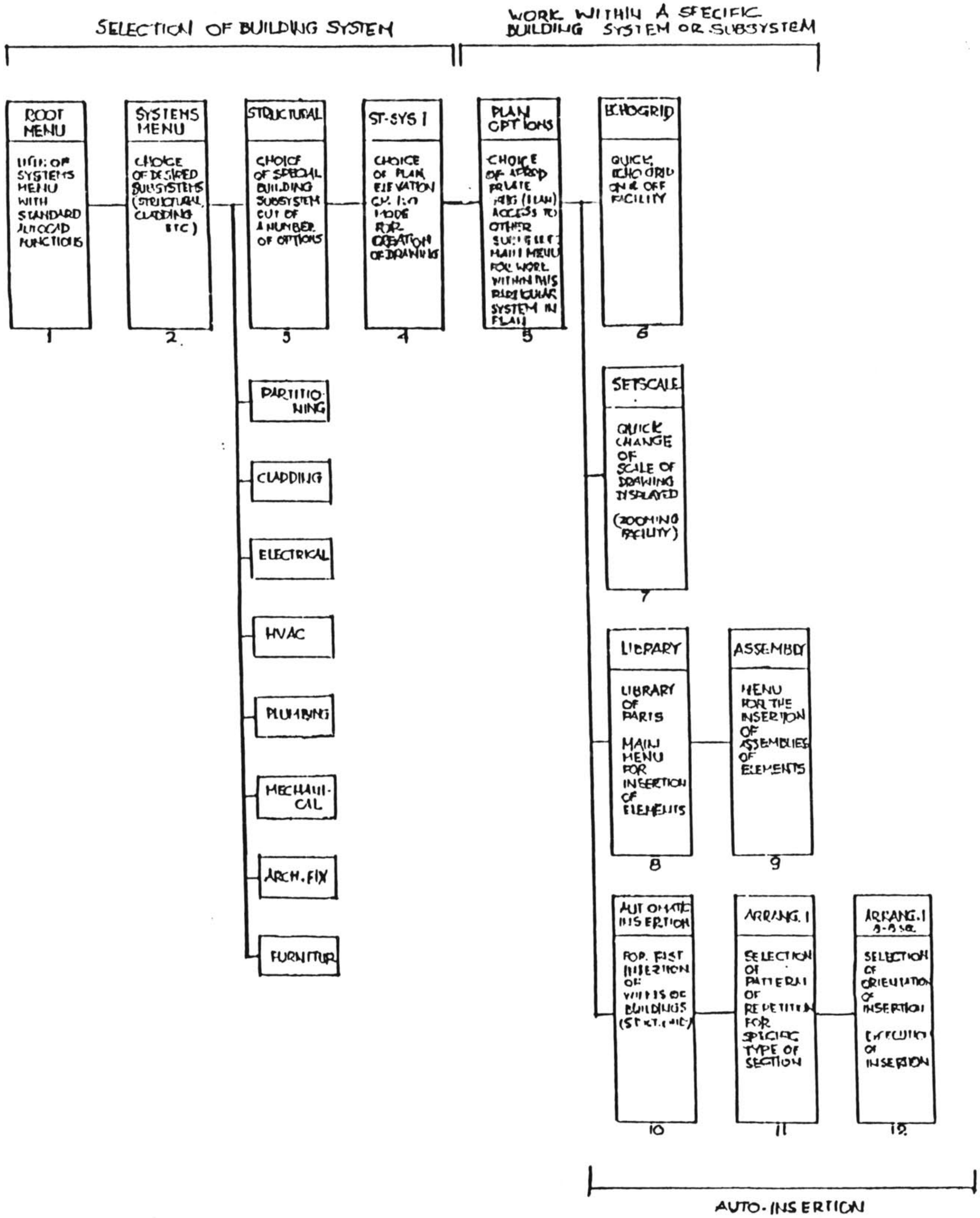
An initial attempt was made to develop such a system by a customization of Autocad version 2.01. This 2-d drafting package offers a wide range of facilities that are of great relevance to the specific application in mind. The spine of the facility is a relatively long custom-made menu file, namely the "systems menu" which is supported by a large number of script files, and several drawings of systemic elements which facilitate the interaction with the system.

The higher levels of the systems menu allow the user to select what subsystem he intends to work with, (structural, partitioning, cladding, electrical, mechanical, hvac, plumbing or AR-fix etc.), to chose the specific subsystem he wants out of a list of available options and to potentially to decide whether he wants to work on plan elevation or isometric mode. However, only the plans can be designed at present since for a two dimentional system the facility that works in elevation is totally independent from that for the plan and the exercise aimed at the indication of principles and design techniques and not the development of a system to be utilized in practice.

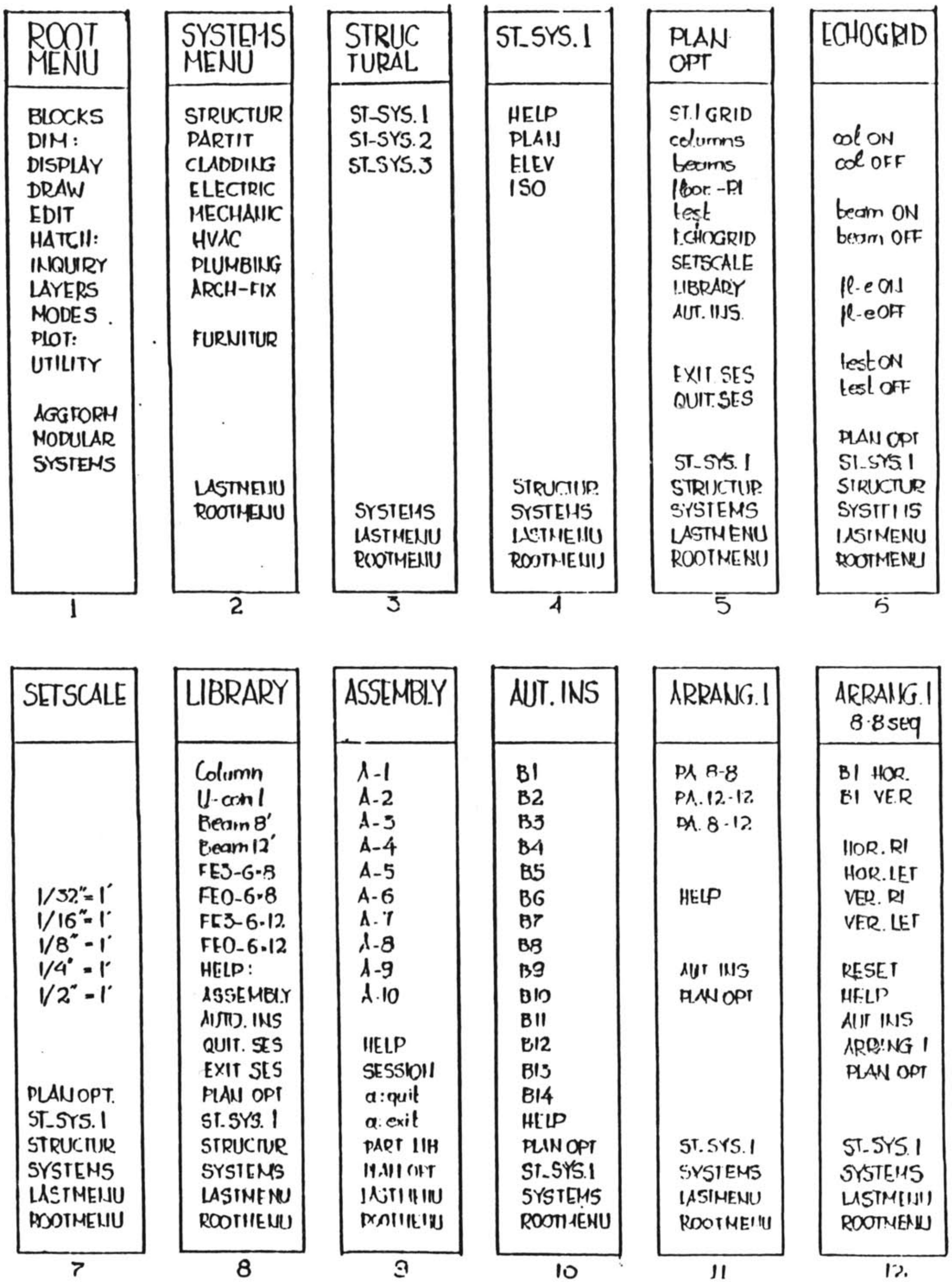
The hierarchical nature of the screen menu seems to be very suitable for this kind of selection process and it facilitates the provision of help on screen at each level of decisions. The lower levels of the screen menu allow for the selection of the appropriate grids that facilitate the insertion of the elements of the specific system chosen, as well as test whether the results are in accordance with the system's rules. There is also facility for quick echoing on and off of the various grids, which are different for each building system or subsystem supported, as well as a facility for rapid change of scale.

The overlay also contains libraries of systemic elements as well as assemblies of these elements. The help on screen at this point becomes vital, providing information of the optional elements or assemblies to be inserted, the location of their insertion base points, rules of insertion, appropriate layer of insertion etc. The various grids are linked with appropriate snap and layering facilities which guarantee the insertion of the systemic elements according to the rules and there is provision for an option for grids that appear on hard-copies and those that only appear on screen.

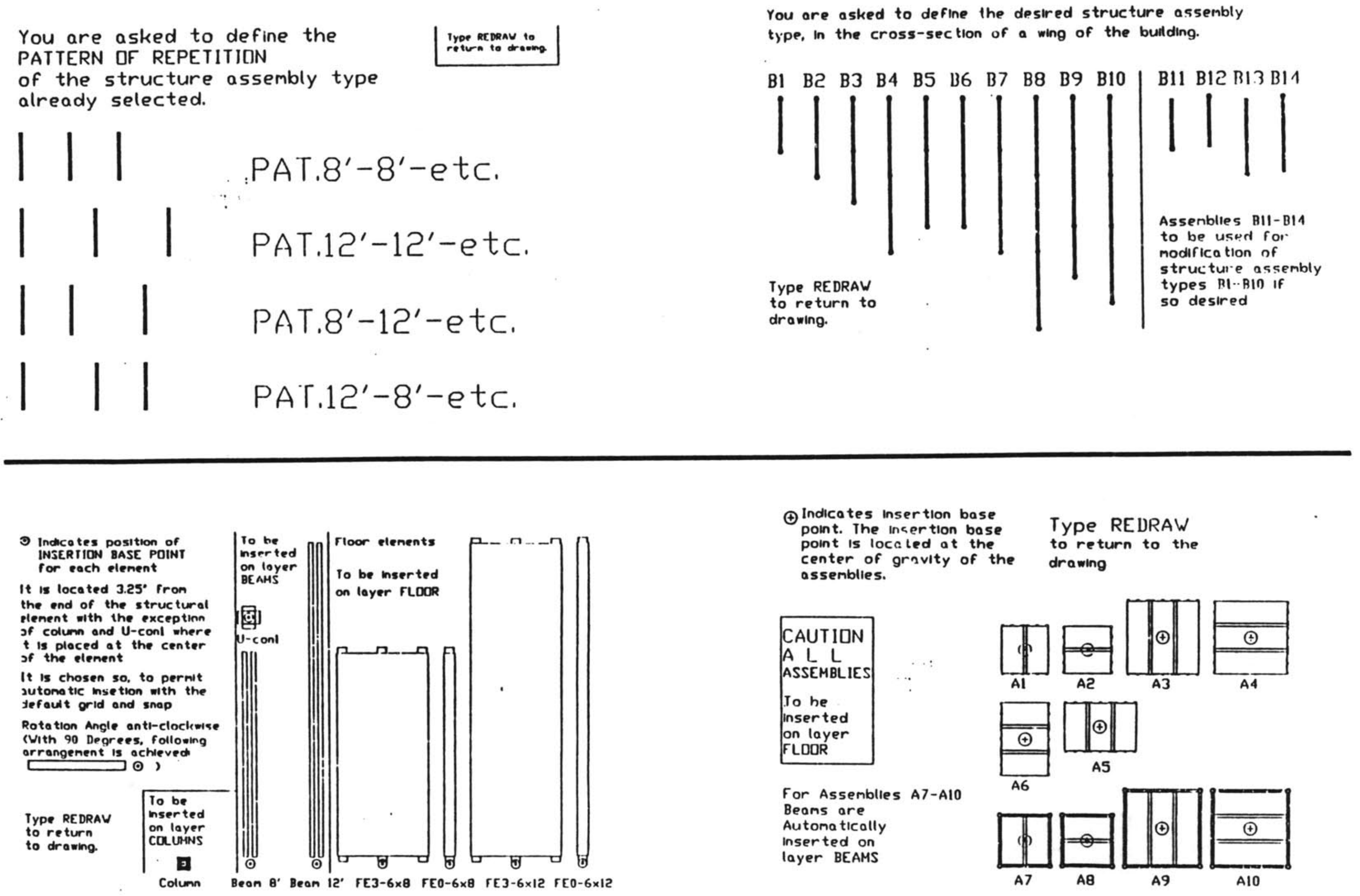
Throughout the use of this facility the standard autocad commands are always accessible to the user either through the screen menu that at a different area can also incorporate the standard autocad menu, or through the tablet menu and keyboard, however it is unlikely that they will ever be needed. Furthermore they should be used with caution and only by experienced users since they might disable some



Diagramatic Scheme of the structure of Module 1.



Sample Screen menus on different levels of Module 1.



You are asked to define the PATTERN OF REPETITION of the structure assembly type already selected.

Type REDRAW to return to drawing.

You are asked to define the desired structure assembly type, in the cross-section of a wing of the building.

Type REDRAW to return to drawing.

Assemblies B11-B14 to be used for modification of structure assembly types B1-B10 if so desired

⊙ Indicates position of INSERTION BASE POINT for each element

It is located 3.25' from the end of the structural element with the exception of column and U-conl where it is placed at the center of the element

It is chosen so, to permit automatic insetion with the default grid and snap

Rotation Angle anti-clockwise (With 90 Degrees, following arrangement is achieved ⊙)

Type REDRAW to return to drawing.

To be inserted on layer COLUMNS

To be inserted on layer BEAMS

Floor elements

To be inserted on layer FLOOR

⊕ Indicates insertion base point. The insertion base point is located at the center of gravity of the assemblies.

Type REDRAW to return to the drawing

CAUTION
A L L
ASSEMBLIES
To be inserted on layer FLOOR

For Assemblies A7-A10 Beams are Automatically inserted on layer BEAMS

Examples of screen menus supporting the structural system. Top : Automatic assembly facility. Bottom left : Library of parts. Bottom right : Assemblies of structural Components suitable for buildings of small scale.

feature of the overlay with unpredictable results.

2.3.2 Features and use of the facility

This package allows the creation of very complex and accurate multi-layered 2D drawings at a very small fraction of the time it would normally be required for a similar job within standard autocad, and what is most important, from relatively unskilled users. The menu has been structured in such a way as to substantially aid in the prevention of typical hazards that are likely to occur when beginners have their first confrontation with a cad package. Such hazards may imply entering information on the wrong layer, using an inappropriate snap value leading to inaccurate drawings or excessive zooming, judging incorrectly the optimum location of the insertion base point of various blocks, etc.

This first overlay on autocad's graphics editor exclusively dedicated to systems building, incorporates a coherent facility for the development of the structural system which offers the user the option of working at two different scales.

The user can either insert components or assemblies of components or alternatively can produce accurate drawings of the structural system of a whole portion -wing- of the building very rapidly by utilizing the automatic insertion facility. This much more efficient method however is not appropriate for generic use since it requires large storage space in order to be maintained which is not justified when the user simply wishes to crate a small floorplan. It is therefore created to assist the user in projects with extensive and complicated layouts. The user is guided through a 3-step selection process from the screen menu and the system automatically inserts the structure for a whole wing of the building in an area defined by the user.

This overlay also incorporates fully developed packages for partitions and cladding. Here work can be performed only in a smaller scale and with a smaller degree of automation, which is due to the fact that the placement of these subsystems involve many design decisions. It was considered that a different approach which would tent to automate this process would be particularly restrictive for the architect and would imply that the system would have to make design decisions based on assumptions that could well be questioned.

The system is capable both for the creation of accurate drawings and production documents given a specific system, and for the testing of the flexibility of the system or comparison with other systems in terms of possible space and furniture arrangements, as well as cost effectiveness of various solutions.

This last feature is to a large extent due to the possibility of linking autocad with a spreadsheet environment (like lotus123) where calculations and analytical work can be rapidly performed in order to test the variety of solutions quickly developed within the system. Therefore this linkage, being of great use to the systems designer that needs to test his system consistently against various criteria and to the project architect that seeks the cheapest and most flexible solution to fulfill his program requirements, is an essential part of this package.

Also for the structural system a different application enabling the automatic insertion of large portions (wings) of the building has been developed mainly to be utilized in the creation of large floorplans. The user is guided through a 3-step selection process from the screen menu and the system automatically inserts the structure for a whole wing of the building in an area defined by the user. This process leads to much faster still production of drawings and is linked directly with the sketch design stage, through standard features of Autocad.

2.3.3 Maintenance of the facility

The systems menu is a relatively long menu file approximately four times as long as the standard Autocad 2.01 menu file.

Since grids for the insertion of elements and test of the rules of the system with the accompanying snap and layering facilities are obviously different for each specific building system it is important to develop a completely different set of screen menus for each subsystem. However the Systems menu is structured in such a way as to allow the quick production of these screen menus within a text editor since they all conform to the same precisely defined rules for their assembly. The same applies for the large number of script files that support the screen menu (approx 350 already developed). The subdivision of tasks among menu file and script files for the automatic execution of series of commands has been decided under the basic consideration of easy update of the files in a text editor.

This initial description is followed by 2 tables, one showing the structure of the systems menu by presenting the basic layers of the screen menu with a short description of the tasks that each one performs, and the other showing the actual menus that appear on the screen.

Until this stage, there has been a lot of experimentation in this project for the potential for economizing both in memory consumption and most important in storage space on the disk, since this issue is considered of major importance on a microcomputer environment. This basically involved utilizing script files for the insertion of assemblies of elements in the drawings instead of following the conventional process of storing the assemblies as drawing blocks in the disk. This eliminates the storage space to approximately one tenth, and this process can be relatively easy to implement in an environment with only a finite number of elements that are produced industrially and assembled in a variety of ways.

The system at present can still fit into a high density floppy disk but access time is getting increasingly slow and soon there will be need to store it onto the harddisk. However still it is essential to have a separate subdirectory for the proper use of this application package that at this point of its development is supported by approx. 410 files.

2.3.4 Hardware and Software utilized

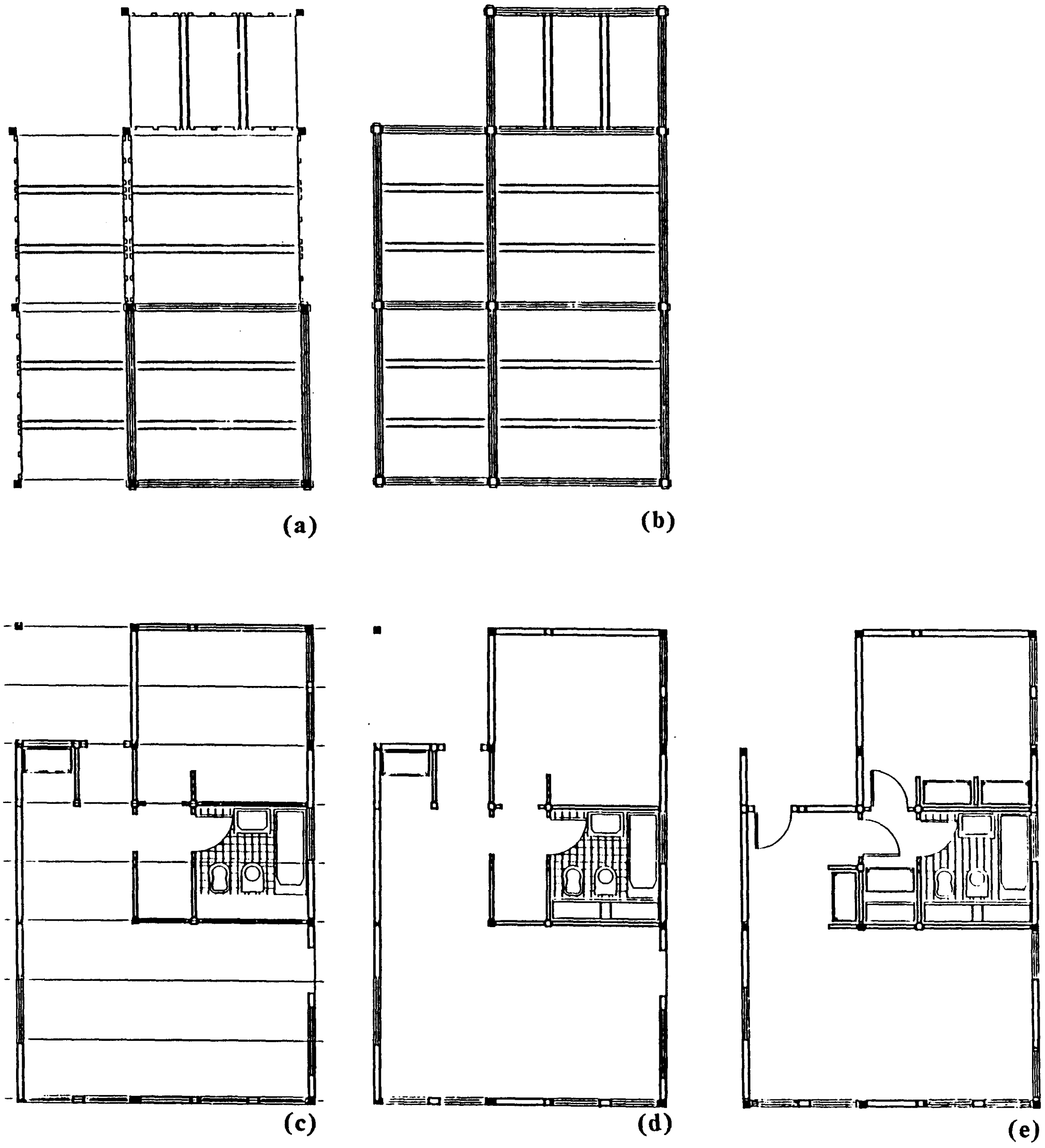
This study was performed with the facilities of the CRL of the department of Architecture and Planning.

Hardware Utilized:

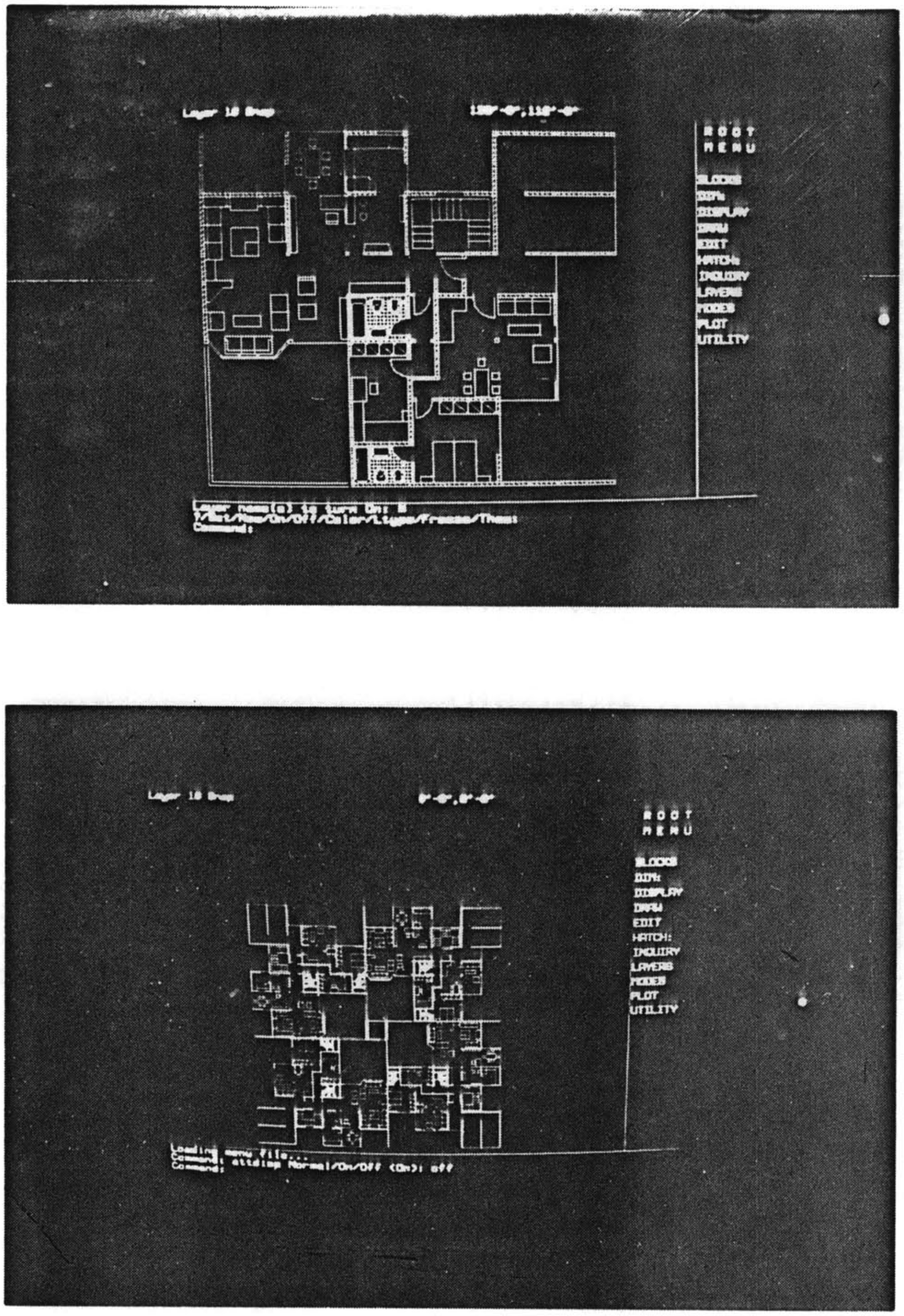
1. An IBM PC-AT (512 K RAM) connected to a monochrome monitor and utilizing a Hercules graphics card.(+2 Asynch. cards)
2. A Summagraphics Bit Pad 2 digitizer.
3. A Plotter. (HP 7475 and later IBM)

Software Utilized:

1. DOS version 3.0 Operating System.
2. Autocad Autodesk Inc. Version 2.01
3. Lotus 123 Lotus Development Inc.



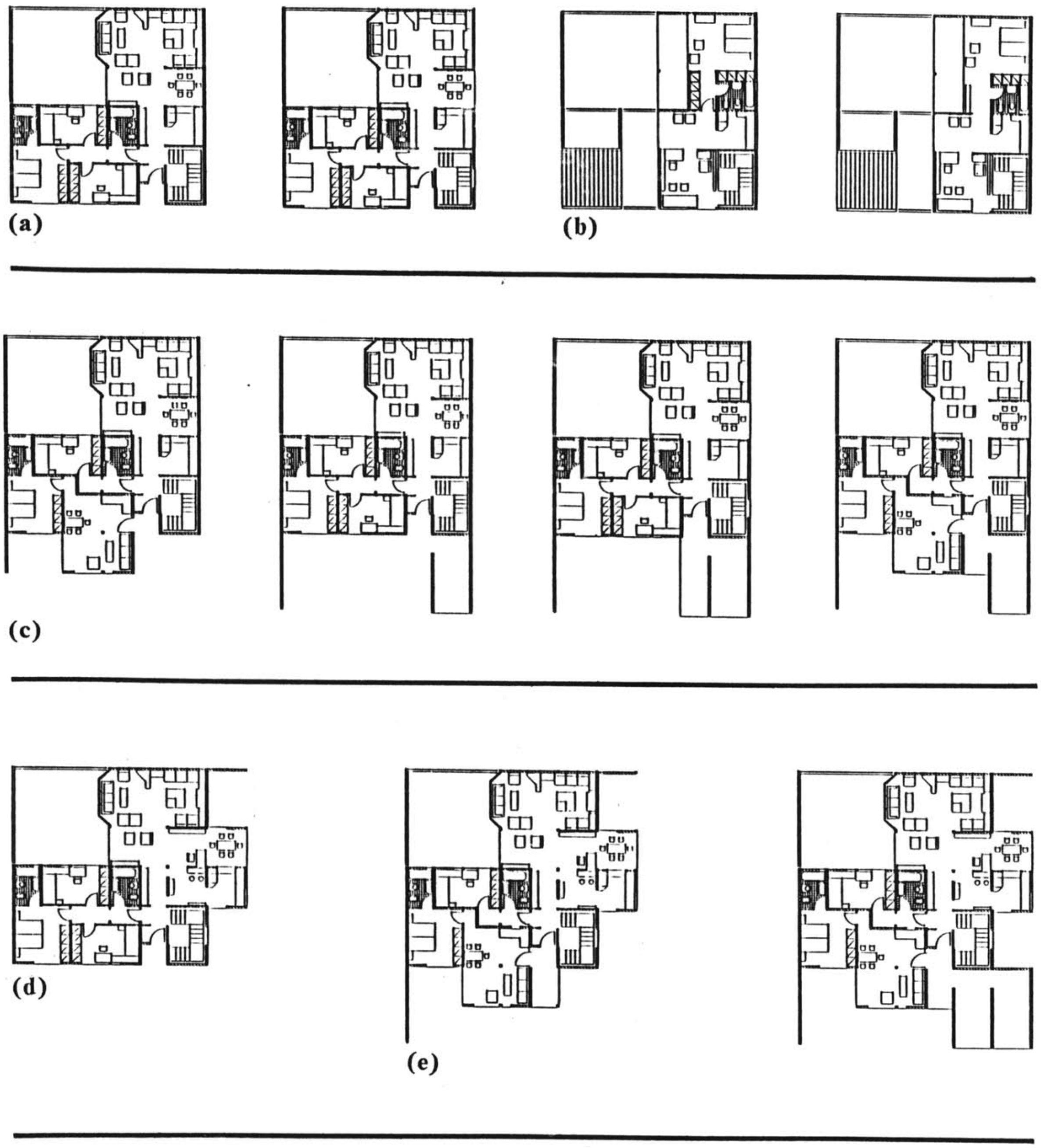
Development of a floor plan by use of the building systems facility (module 1). (a) Developing of the structural system. (b) The structural system complete. (c) Use of auxiliary grid for the use of cladding and partitioning subsystems. (d) Placement of the HVAC and Mechanical Core. (e) Placement of fixed elements completed.



```
Layer name(s) to turn On: B
?/Set/New/On/Off/Color/Ltype/Freeze/Thaw:
Command:
```

```
Loading menu file...
Command: attdisp Normal/On/Off <On>: off
Command:
```

Utilization of Module 1. to test suitability of Building System for Low-rise Courtyard housing. (a) Typical corner unit floor plan. (b) Four unit cluster.



Few of the floor plan variations developed by the system. (a) Minimum unit. (b) Second storey variations. (c) Expansion in one direction only. (d) Expansion in the opposite direction (e) Expansion in both directions - Corner units.

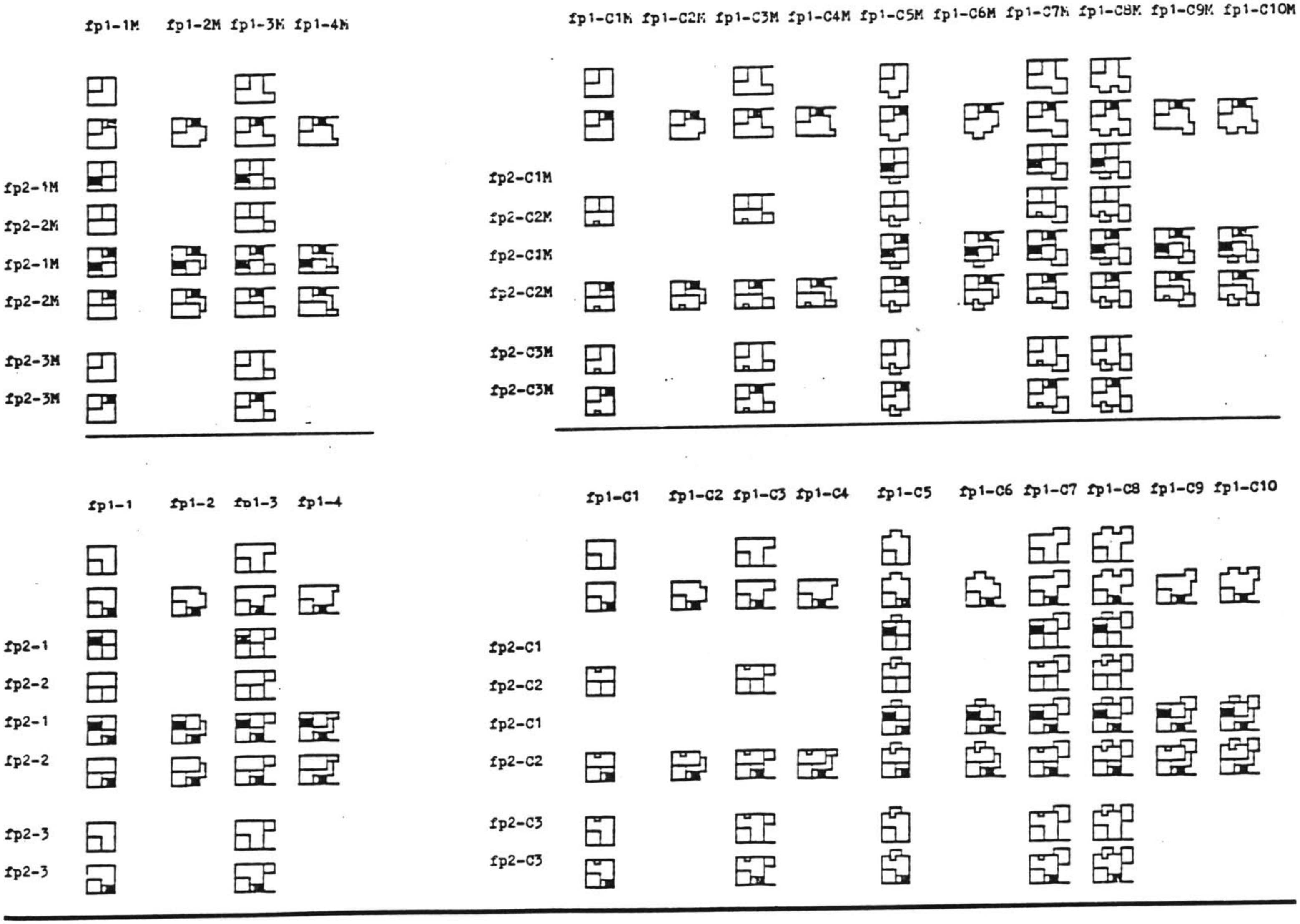
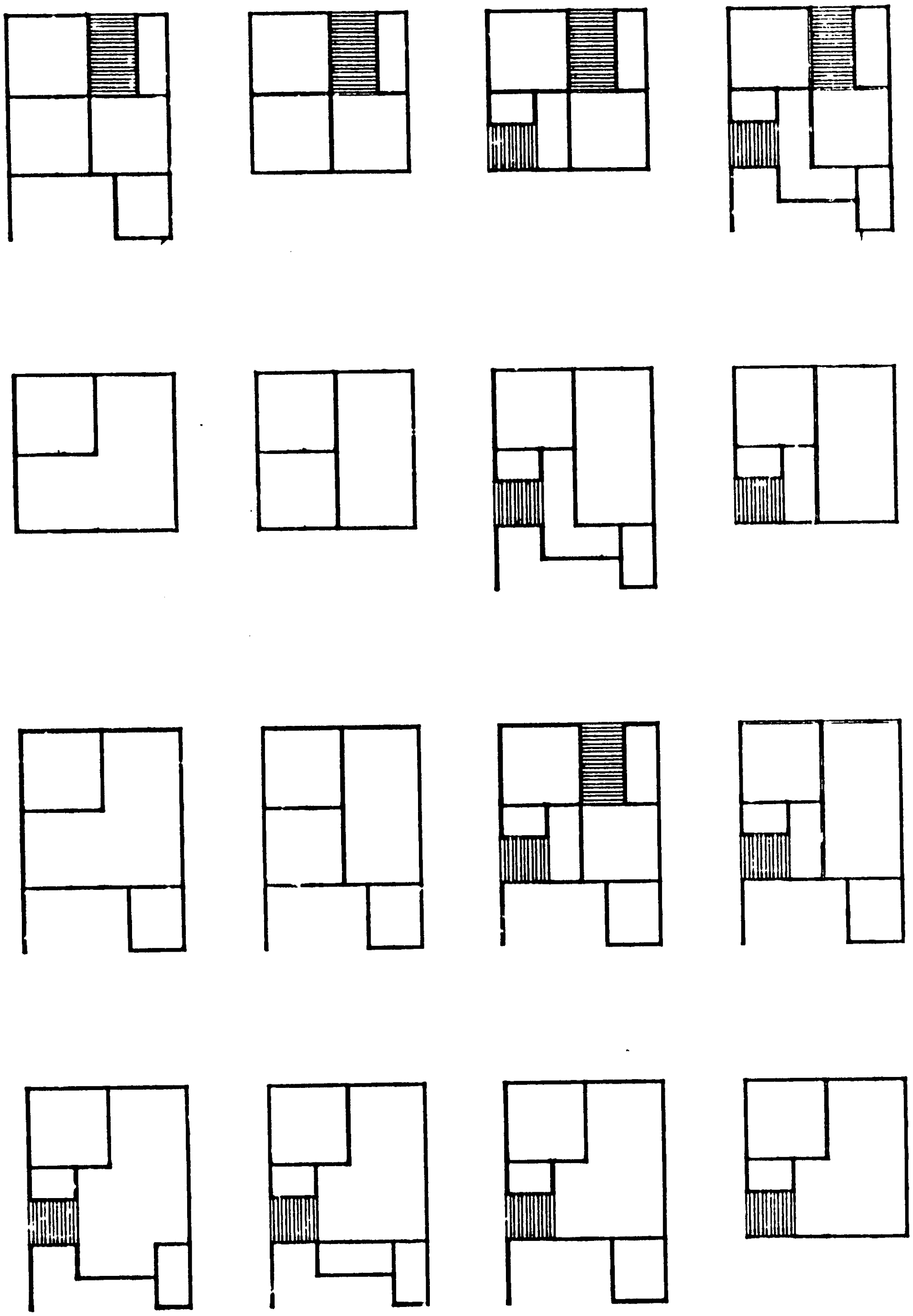


Table of potential volumetric variations within the given site, associated with alternative floor plans.



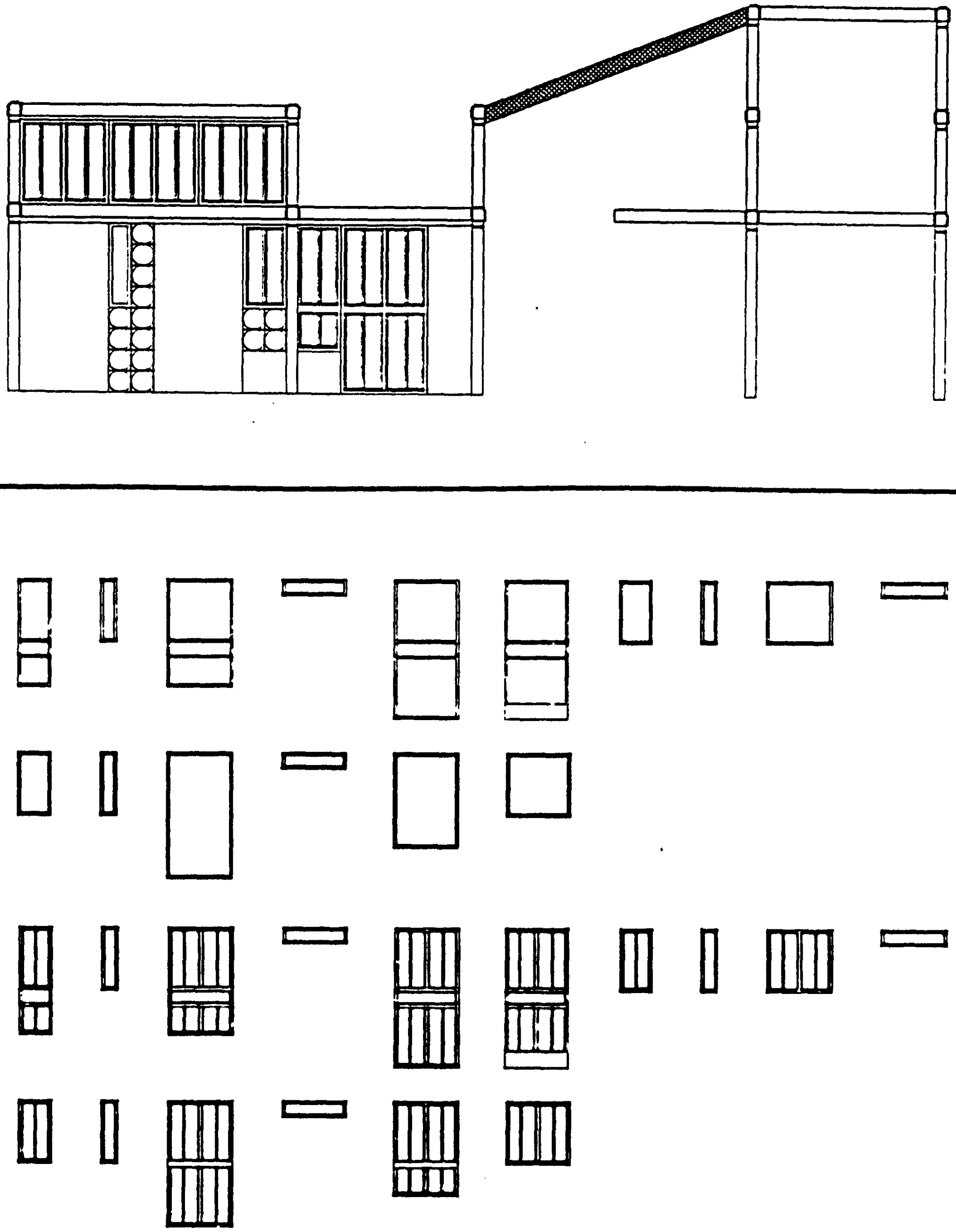
Detail of few of the roof plan variations developed, indicating volumetric flexibility of the system.

```
CODE    LENGTH  HEIGHT  THICKNESS   COST

P2-4    0.6     2       0.1         12
P2-4    0.6     2       0.1         12
P2-4    0.6     2       0.1         12
P2-4    0.6     2       0.1         12
P2-4    0.6     2       0.1         12
P2-4    0.6     2       0.1         12
P2-4    0.6     2       0.1         12
P2-4    0.6     2       0.1         12
P2-4    0.6     2       0.1         12

CODE    LENGTH  HEIGHT  THICKNESS   COST
p1-4    0.3     2       0.1         6

CODE    LENGTH  HEIGHT  THICKNESS   COST
p1/2-4  0.15    2       0.1         3
p1/2-4  0.15    2       0.1         3
p1/2-4  0.15    2       0.1         3
p1/2-4  0.15    2       0.1         3
p1/2-4  0.15    2       0.1         3
p1/2-4  0.15    2       0.1         3
p1/2-4  0.15    2       0.1         3

CODE    LENGTH  HEIGHT  THICKNESS   COST
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
p4-4    1.2     2       0.1         24
```

```
CODE    LENGTH  HEIGHT  THICKNESS   COST

P2-6    0.6     2       0.15        24
P2-6    0.6     2       0.15        24
P2-6    0.6     2       0.15        24
P2-6    0.6     2       0.15        24
P2-6    0.6     2       0.15        24
P2-6    0.6     2       0.15        24
P2-6    0.6     2       0.15        24
P2-6    0.6     2       0.15        24
P2-6    0.6     2       0.15        24
P2-6    0.6     2       0.15        24
P2-6    0.6     2       0.15        24
P2-6    0.6     2       0.15        24
CODE    LENGTH  HEIGHT  THICKNESS   COST
p1-6    0.3     2       0.15        12
p1-6    0.3     2       0.15        12
p1-6    0.3     2       0.15        12
p1-6    0.3     2       0.15        12

CODE    LENGTH  HEIGHT  THICKNESS   COST
p1/2-6  0.15    2       0.15        6
p1/2-6  0.15    2       0.15        6
p1/2-6  0.15    2       0.15        6
p1/2-6  0.15    2       0.15        6
p1/2-6  0.15    2       0.15        6
p1/2-6  0.15    2       0.15        6
p1/2-6  0.15    2       0.15        6

CODE    LENGTH  HEIGHT  THICKNESS   COST
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48

CODE    LENGTH  HEIGHT  THICKNESS   COST
p4-6    1.2     2       0.15        48
p4-6    1.2     2       0.15        48
```

```
                                           HISTOGRAM
NUMBER OF COMPONENT TYPE 1 (P2-4)            9        TOTAL COST OF COMPONENT TYPE 1     108
NUMBER OF COMPONENT TYPE 2 (P2-6)           12        TOTAL COST OF COMPONENT TYPE 2     288
NUMBER OF COMPONENT TYPE 3 (P1-4)            1        TOTAL COST OF COMPONENT TYPE 3       6
NUMBER OF COMPONENT TYPE 4 (P1-6)            4        TOTAL COST OF COMPONENT TYPE 4      48
NUMBER OF COMPONENT TYPE 5 (P1/2-4)          7        TOTAL COST OF COMPONENT TYPE 5      21
NUMBER OF COMPONENT TYPE 6 (P1/2-6)          7        TOTAL COST OF COMPONENT TYPE 6      42
NUMBER OF COMPONENT TYPE 7 (P4-4)           13        TOTAL COST OF COMPONENT TYPE 7     312
NUMBER OF COMPONENT TYPE 8 (P4-6)           51        TOTAL COST OF COMPONENT TYPE 8    2448

TOTAL NUMBER OF COMPONENTS                 104        TOTAL COST OF COMPONENTS          3273
```

Example of the output extracted from the graphics editor, which is processed in a spreadsheet environment.



Top : A section of the structural system showing part of facade. Bottom : Alternative sets of systemic cladding components.

3. COMPONENT BUILDING - MODULE 2

3.1 Introduction

This work is an overlay on AUTOCAD'S graphics editor that makes the editor a suitable tool for the rapid creation of a 3D building description, when utilizing a systems approach in building.

The development of this tool was based on a clearly defined set of user requirements described earlier in this document in section 2. It is a facility that can be used by a project architect in the design of a project which is going to be built with a previously selected building system. The goal is to simplify his task without converting him from his customary process of designing.

In the description of this facility which will follow, design concepts already mentioned in the previous module will not be discussed again. This package has been developed based on the experience gained by the previous module, it is a continuation of the same experiment within a more mature and friendly environment and can be linked directly to the previous module. The document that describes the facility is structured similarly and the reader is therefore assumed to have some insight on the arguments presented in the description of the previous module.

This overlay indicates the advantages of having a powerful programming environment for the customization of the graphics editor for specific uses, and the potential of LISP as a programming language in this process.

3.1.1 The programming environment

This work is a continuation of the first module for component building which was described previously in this document. Although both packages - overlays on AUTOCAD'S graphics editor are independent they represent a continuous effort towards the same goal and are of particular interest when considered as a sequence.

This need to view these modules as a sequence is indeed justified not only by the fact that the development of the second module was based on the experience acquired by the development of the first but also by the fact that the result reflects to a large extent the significant changes in the environment within which these modules were programmed.

Indeed as already stated previously, the impact of microcomputers in the field of architecture has been considerable and this environment has been evolving fast. The changes in this environment over the last year have been impressive both in terms of hardware developments and in terms of software that runs on the microcomputers which is of relevance to architectural practice. The Computer Resource Lab of the School of Architecture and Planning here at M.I.T. which until recently has been exclusively microcomputer based, was a very appropriate environment for one to anticipate these changes.

Thus, the second module, although developed only a few months later than the first was programmed in a much more powerful environment. The generic graphics editor, AUTOCAD, on which this second module is too an overlay, was now versions 2.15 and 2.17f which both contained 3D features and a new LISP interface for internal programming and customization.

Both these new facilities were far from perfect. Autocad 2.17f is not yet a true 3D package, can only perform extrusions from the XY plane, does not have capability for perspective, and incorporates a particularly slow hidden line capability. The LISP interface (AUTOLISP) on the other hand, is currently being debugged, has still several problems related to memory allocation which enforce a rather poor programming style which is closer to linear programming and does not enable one to take fully advantage of the power of common LISP.

Furthermore there are several limitations preventing access to AUTOCAD'S database and attribute information once initially stored in a drawing are practically "lost" since there is no way to interrogate the data base about it, to select a group of entities by attribute value etc. The only way to access attribute information once stored within a drawing is to extract it and manipulate it outside. This however is a serious limitation when attempting to customize the graphics editor through internal programming.

However the improvement from versions 2.00 and 2.01 is remarkable. Furthermore it seems that efforts of the software vendor are concentrated towards the right direction. From a practically non existent programming environment in the previous versions there is now a LISP interface allowing programming in common LISP with the addition of several autocad specific commands and facilities regardless of the few still existing serious limitations, as well as a wide range of system variables which can be accessed by the programmer. From a 2D representation we have moved to a primitive, but particularly useful 3D facility which with several restrictions and a lot of strain from the user's side can potentially create a 3D representation of the building.

Within a few months the attempt to internally program the graphics editor in a microcomputer environment in order to build a set of overlays that would make it a flexible and multi-purpose tool was no longer a risky experiment but a potentially realizable goal. Several software vendors already started to market such overlays with success.

If this progress continues with the same success both from graphics editor suppliers and from individuals developing overlays for these editors, the notion of the modular graphics editor and the open computer system for architectural practice advocated in this thesis will soon be an obvious reality.

3.1.2 User requirements - Goals

The primary distinction from the previous module developed in the spring, is that this package is focused on the potential for use of the facility by the project architect in professional practice. It is therefore based on a set of assumptions about a probable scenario for the use of the system and aspires to efficiently utilize innovative techniques that will make its use consistent with a desirable design process.

It demonstrates the potential of building a tool which guarantees that the incorporation of a particular subsystem in the design of a building shall facilitate the job of the project architect. This tool should allow the designer to become easily aware of the rules of the system without forcing him to abandon his customary and preferred design process. It also helps him produce the required construction documentation with the minimum possible effort.

This facility is an overlay on a limited by nevertheless 3D graphics editor and therefore has the potential of creating a 3D representation of the building instead of being limited to building 2D drawings. This fact alone makes it a much more powerful tool for the application of systems building techniques in architectural practice.

The requirement of making it an useful facility for architectural practice implies primarily the need to utilize the 3D capability of the graphics editor and to force decisions to be taken as early as possible in the design process. This leads inevitably to the development of a powerful facility for schematic design as well as a structure of the menu which provides the potential to evaluate the efficiency of the solution at several stages of the design process. Obviously different sets of evaluation packages have to be developed for each stage of the design process.

This project was not restricted by a given building system. Its goal was to indicate several design principles for a C.B.B.S without accepting the limitations imposed by pre-existing building systems. The goal is not the final product which requires much more research and a far better still programming environment and generic graphics editor in order to be efficient. Instead the purpose of this exercise is to indicate the utility of some of the ideas presented previously in the document and to illustrate the potential of already existing facilities and the resulting improvement from previous work.

3.2 Significance of the project - Principles and Techniques indicated

3.2.1 Significance in a conceptual level

The development of this facility provided particularly useful insights on the potential for development of an overlay for systems building to be used in architectural practice. Its significance at the conceptual level can be summarized as follows:

- This overlay attempts to allow the introduction of systems building techniques in architectural practice without converting the practicing architects from their already developed process of designing.

- The overlay is conceptually divided into three clearly defined parts. One provides the user with information about the building system before he starts to utilize the system for the development of the building description, the next guides him through the schematic design process and helps him develop and select an optimal solution, and the third is involved in the final design. There is also a facility for the production of construction documentation.

- Decisions are taken as early as possible in the process. Top-down design possible. Clear distinction between the kinds of decisions that can be taken during the schematic design stage and the final design stage.

- The overlay is structured in a way that allows the utilization of separate sets of analyses packages at each stage.

- It indicates advantages and potential of LISP environment for internal programming.

- It includes techniques for easy manipulation of a building description of a reduced size created by the insertion of simplified components, which by a process of automatic redefinition can produce the final building description with the complete set of 3D components.

3.2.2 Significance in a practical level

- This facility is of particular use to a project architect. The goal is to make the process of designing a building to be built with a particular pre-selected set of industrially produced building subsystems very similar but a lot easier than designing a traditional building. This tool is thus aiming in the facilitation of the application of systems building techniques in architecture.

- This tool due to its 3D capability as well as techniques for the transition among different subsystems is particularly sensitive in the interfaces among various subsystems.

- The overlay incorporates a well developed schematic design facility and has separate facilities for the final design and the production of construction documentation. Accurate 2D and 3D drawings can be rapidly and efficiently produced.

- Advanced and versatile techniques for the display of the rules of the system both during the design process and through on the screen tutorial. This part also to be used for marketing of the product, in schools, etc.

3.2.3 Design principles and techniques indicated

The practical utility of the particular project has already been described. It is important however to focus on the some of the design strategies it illustrates :

The Potential offered by breaking up the menu into smaller more manageable units. These menus are structured in a hierarchical way in such a way that the user does not notice the fact that spine of the facility is not one but several menus. This technique allows easier management of the whole project and prevents excessive nesting within a given menu which is both undesirable and often presents technical problems.

Based on the previously described fragmentation of the menu and by placing a top-level menu which is responsible for the appropriate initial drawing setup the user has an option of saving time by utilizing a different initial menu according to whether he wishes to enter an existing drawing or create a new one. Furthermore this arrangement provides the option of directing the user to a different set of applications according to the selected size and scale of his drawing which determines the scale of his project. This is a more efficient way to implement the principle of utilizing different techniques for the use of a particular building subsystem according to the scale of the project as described in the previous module for component building.

An other important technique that is demonstrated in this module is related to savings in drawing time and response time on the micro - workstation by redefining components. This technique allows a drawing to be kept relatively small throughout the design process and therefore to be manipulated with ease. When and if detailed drawings are required, these can be produced automatically upon request at the end of the design process. This automated process is part of the construction documentation phase.

Ideally alphanumeric information is stored in the drawing within the set of simple components that the user manipulates throughout the design process, and from this information which is stored in the form of attributes, the various analyses packages are run outside the environment of the graphics editor. The set of final detailed components do not necessarily need to be associated with attribute information.

The project also indicates a method for interactive definition of attributes during the design process which saves the user considerable amount of time. The user during the design process makes several selections (e.g. regarding the type of a brick wall) which are stored in variables. The definition of these variables is not only responsible for the prompts the user will receive by the system but will also automatically define the values of the various properties of the attributes stored within the systemic part to be inserted.

This advancement saves storage space since there is no longer need for several drawing files with constant attributes representing variations of the same component type to be explicitly stored. Furthermore it permits the user to insert components, of any size allowable by the rules of the building system, by utilizing a single prototype drawing file. The user is also no longer prompted explicitly for the attribute values of his component.

3.3 Detailed description of the facility

3.3.1 Design of the facility - Structure of the menu

This facility was developed on AUTOCAD 2.17f. This, still not fully 3D package, is primarily oriented towards drafting and applications for mechanical engineers, but with the LISP interface it contains for internal programming as well as the good communication with the environment around it, proves very useful as a generic graphics editor to be further customized, and is probably the most appropriate package for this purpose available in the microcomputer market today.

In this module, four menus were developed. The menu at the highest level performs the initial drawing setup and is used optionally when a new drawing is created. In the second level is the systems menu which contains the structural system with a schematic design capability as well as a final design facility and a facility for the production of construction documentation. In the lowest level two menus have been developed for the description of two alternative partitioning subsystems which will be utilized during the detailed design stage.

Normally each menu contains information about a given subsystem. However at the top level the menus are structured differently. The highest level is responsible for the initial drawing setup while the level below is dedicated primarily to schematic design facilities which are developed by utilizing the rules of a certain structural system. This technique allows the user to select a different initial menu according to whether he wishes to edit an existing drawing or whether he wants to create a new one. This selection process can be automated by an appropriate setup script.

3.3.2 Features and use of the facility

This package is designed according to the generic set of user requirements described in section 2 of this document. It is meant to be the drawing resident part of the C.B.I.S. and it aspires to meet these requirements to the best possible extent despite the still serious limitations of both the graphics editor and the programming environment provided within it. It does not aspire to be considered a fully developed final product but was developed in order to indicate the design principles advocated previously in this document.

It incorporates features that allow the user to learn about the building system and its rules (tutorial), to develop rapidly alternative design layouts and potentially to evaluate them (schematic design phase), to perform the final design and to produce construction documentation. These features of the system shall be explicitly described below.

A complete tutorial was developed for one of the partitioning subsystems. It contains predetermined slide shows for different topics which can be selected by the user from the screen menu according to his needs, as well as a glossary and an enhancement of AUTOCAD's help facility for the specific application. The slides contain both text and graphic information as already established from the previous module. A large portion of the slides has been numbered in a way that allows the user to also view slides in several different modes related to speed, backwards and forward, or to request to view a specific slide.

This turorial can be used either for marketing purposes and advertizing of the building system or subsystem, for the display of the rules and utility of the system to the project architect in order for him to decide whether it is appropriate to his needs, or for simply conveying information about the rules of the system in order to ensure some familiarity of the designer with the system. In this manner it can also be utilized upon request during the interactive design of the project in order to clarify some principle or rule of the system.

A facility for schematic design of the system was also developed, obviously linked with the structural system and its rules of assembly. It allows the user to rapidly develop alternative project layouts containing (built 3D) volumes which are compatible with the rules of the particular structural system selected. The user can then view these alternative layouts in 3D and evaluate these solutions in several ways. This scenario enables the use of analysis subroutines at various stages of the design process.

The facility for detailed design in which the user can easily create the final building description containing the actual set of structural elements appropriately placed. At this level the user is likely to create a drawing containing detailed 2D images of the structural components. These contain attribute information and can be used for the production of floor plans as well as for further reference in the database and the interfaces with the other subsystems production of reports etc.

The process of creating the detailed building description of the structural system is automated and very similar to the automatic insertion facility of module 1. However it is not complete and can only insert wings of one predefined span since its purpose is to indicate the potential for link with the schematic design phase. It is entirely reprogrammed in AUTOLISP.

The facility for selection among compatible infill subsystems allows the user to decide which subsystem is more appropriate for his needs and directs him according to his decision to the appropriate menu for the partitioning subsystem.

Before the user starts to work with the infill subsystem the drawing is appropriately automatically converted in order for it to be suitable for further stages of the final design with the selected infill subsystem (interior partitioning and cladding subsystems).

Two alternative infill subsystems are been provided at this point. Both are based on different design principles for the insertion of the systemic components as well as the technique for the definition of attributes. One of the system inserts components from a library of parts with predefined attributes while the other creates the components during the insertion by transforming a generic file and inserting a template drawing containing predefined variable attributes that are given a value automatically during the insertion process.

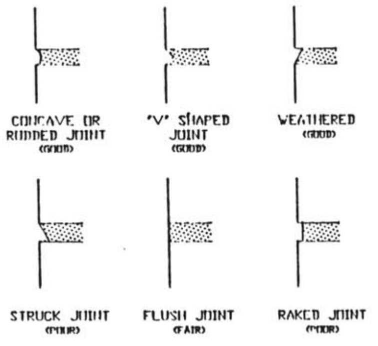
The module is also equipped with a facility for production of construction documentation for the structural system including the automatic replacement of the components with 3D components in a new drawing and the development of final presentation drawings if so required.

3.3.3 Maintenance of the facility

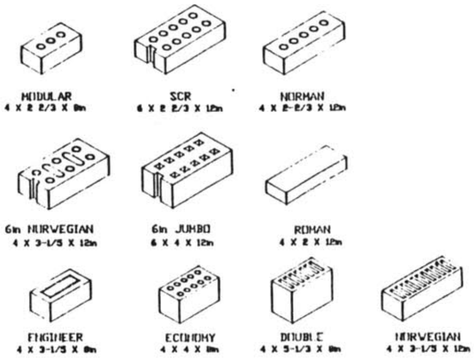
The second module for component building is even easier to maintain than the first one for two reasons. First, the much more powerful programming environment allowed for the development of an efficient tool with the need of much less code both in terms of the size of the menu and the number of the files that support it.

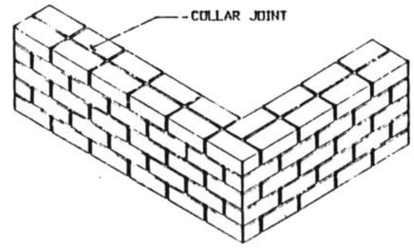
Most important however the weight of the program has shifted from the menu and the script files, to AUTOLISP files. Since AUTOLISP is a version of common LISP with a few autocad specific commands it is obvious that the code is more generic and most unlikely to need to be updated frequently than the script files that depended exclusively on the internal syntax of the AUTOCAD commands.

To this, one must add that the drawing files too, which support the program are considerably fewer than those of module 1. This has to do with the more sophisticated techniques for the insertion of attributes which were enabled by the improved programming environment but also by the fact that the automatic insertion facility which required a lot of support from drawing files was not included in full in the second module but only to the extent that it demonstrated the principle and potential to be linked with other parts of the overlay.

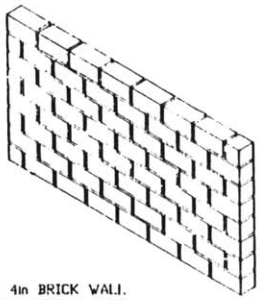


TYPES OF JOINTS (WEATHERABILITY)





8in DOUBLE WYTHE MODULAR BRICK WALL CONSTRUCTION LAYOUT



4in BRICK WALL

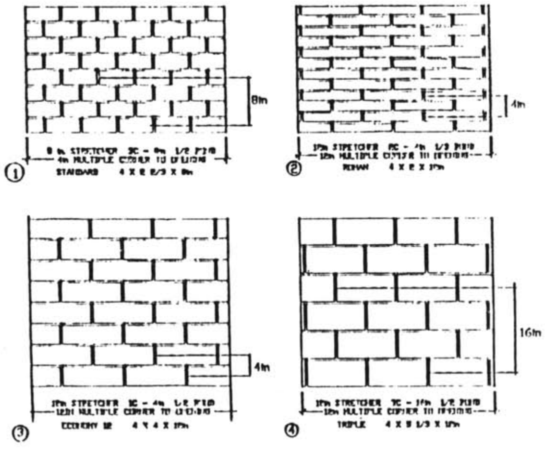
RULES AND USES FOR 4in SINGLE WYTHE BRICK WALLS

NON - LOADBEARING WALL

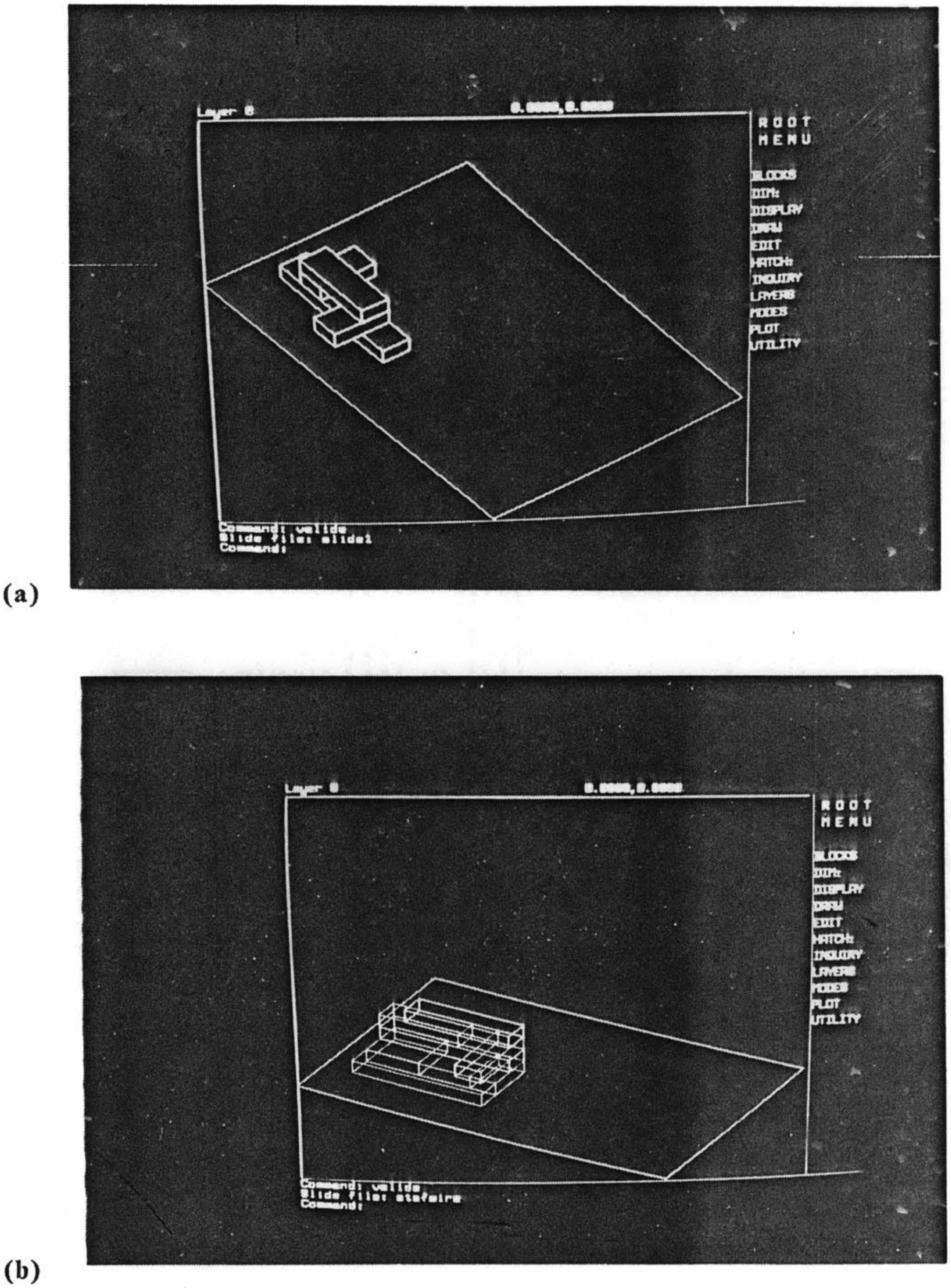
CURTAIN OR PANEL WALL

BRICK VENEER

HORIZONTAL JOINT REINFORCEMENT TO CONTROL SHRINKAGE CRACKS SPACED 16in O.C. VERTICALLY (SEE DETAIL ___)



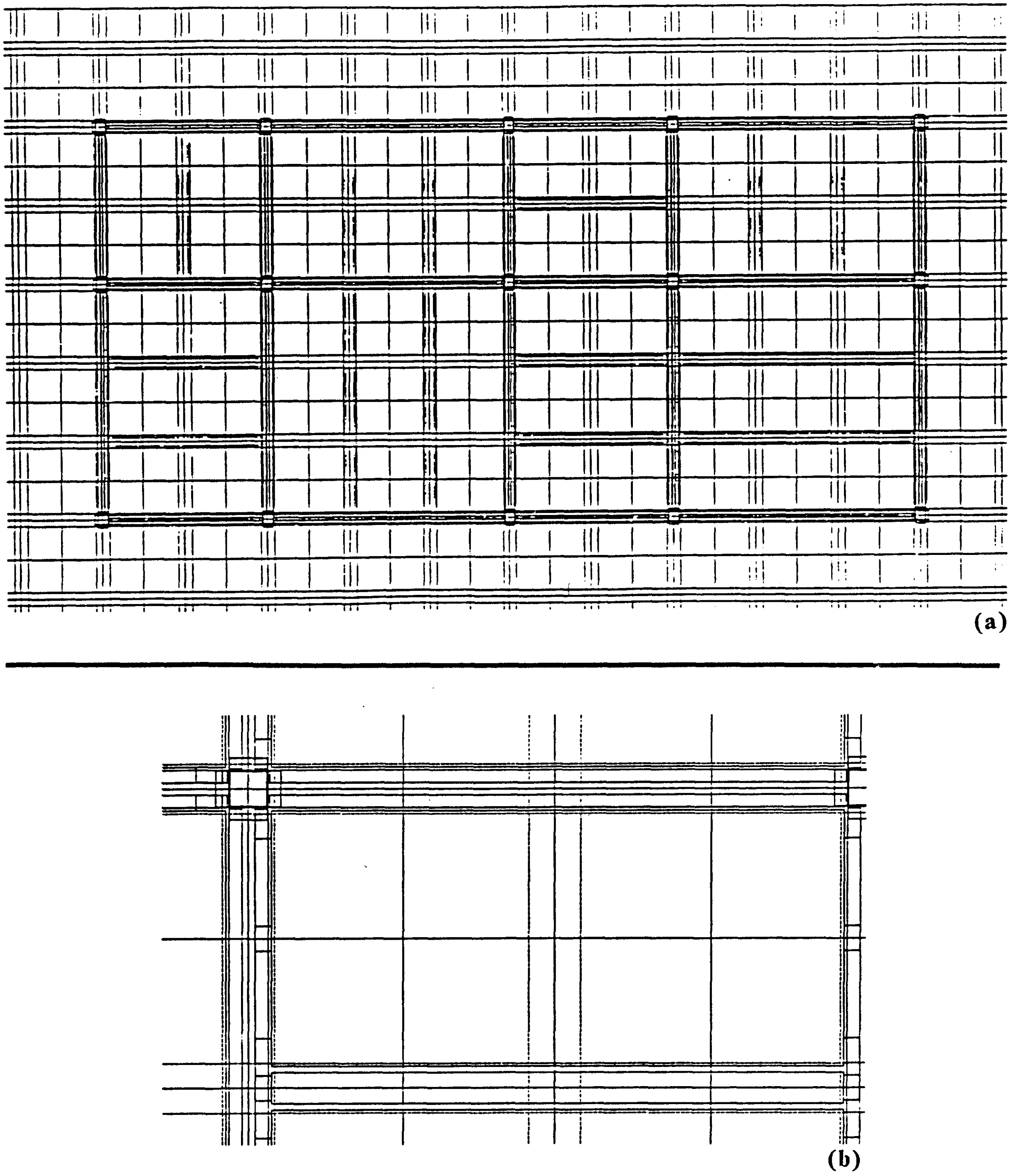
Illustrations on slides from the interactive tutorial and the help on screen from the second (brick) partitioning subsystem.



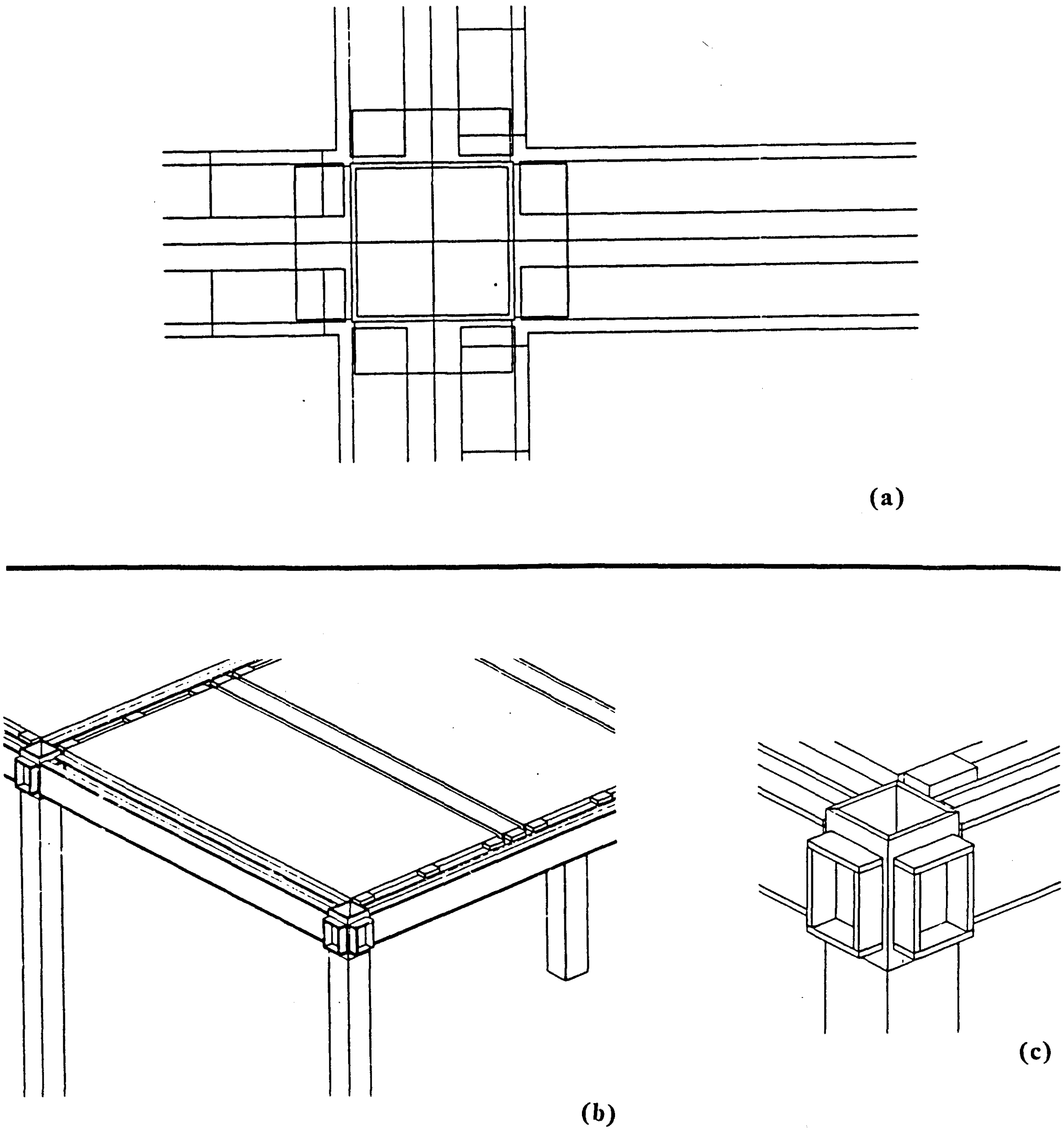
```
Command: vslide
Slide file: slide1
Command:
```

```
Command: vslide
Slide file: stafeirs
Command:
```

The skematic design phase. Use of module 2 to design a building according to the rules of a structural system. (a) Hidden line removal (b) Wire frame. The actual drawings are multicolored according to the floor the volume is inserted in.



Detailed design of the structural system. The user selects the volumes and the structural system is inserted automatically. (a) Structural system layout with 8 and 12 foot spans. (b) Detail showing auxiliary grids and floor elements.



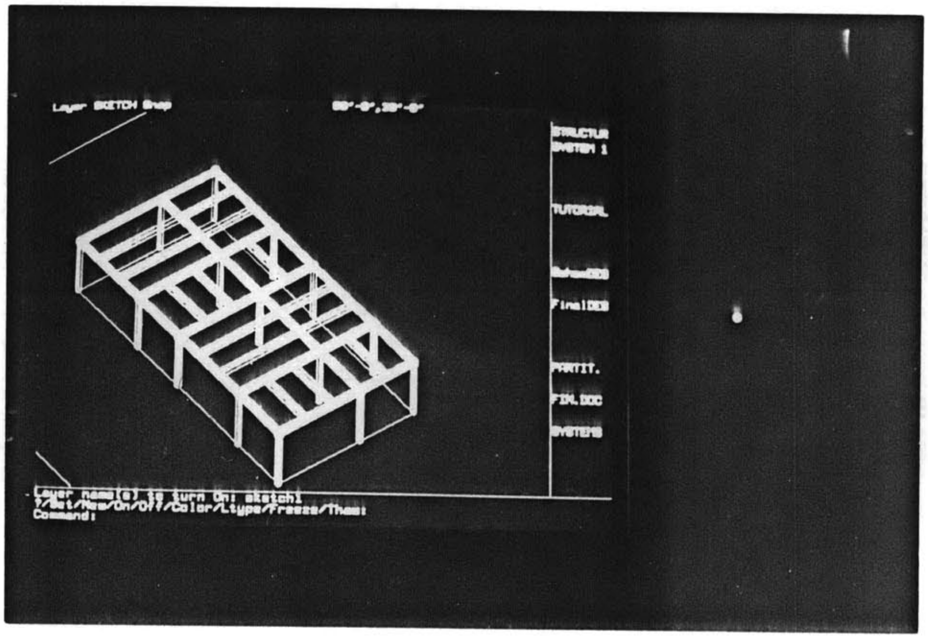
(a)

(b)

(c)

Details of structural system. Production of construction documentation stage. (a) Junction of universal connector, beams and floor elements, in floor plan without hidden line removal. (b) Axonometric of the structural system after the redefinition of the 2-D structural components and their replacement with 3-D components. - Hidden lines removed. (c) Detail of universal connector - Hidden lines removed.

The hardware and software utilized for this application is similar to that of module 1. and will not be explicitly mentioned here. AUTOCAD in this case is version 2.17f instead of 2.01.



```
Layer name(s) to turn On: sketch1
?/Set/New/On/Off/Color/Ltype/Freeze/Thaw:
Command:
```

Automatic replacement of volume indicated by a box during the schematic design stage, with a complete representation of the structural system.

4. DESIGN FACILITY FOR NON SYSTEMIC BUILDING MODULES

4.1 Introduction

This work is an overlay on Autocads' graphics editor that makes it a suitable environment for the creation of floorplans for projects where the user does not wish to be restricted with constraints related to standard widths of walls or angle for their placement.

This could well be the case with a project where the architect is asked to modify an already existing building or operate within a vernacular environment. In this case it is likely that a large number of walls are not parallel to an orthogonal system, do not intersect in right angles, are all of different width and contain openings, windows and doors which are custom made and each one of different size and type.

In this environment, it is obvious that all constraints imposed by most graphics editors upon their users in exchange with speed and efficiency are not relevant. However this does not imply that the architect has to go through a very lengthy process for the insertion of each individual entity of the drawing, or through very complicated procedures or a trial and error process, that will allow him to calculate or eventually approximate the right position and appropriate size and rotation angle of a door or a window.

Indeed this facility proves that even when working within a microcomputer environment which still imposes several limitations related to speed, memory etc.,it is possible to develop a tool that is both flexible and efficient and performs such operations in a generic way and fast.

This facility is a very important addition to a general computational environment for architectural practice. It is a very generic facility that can be used in several applications by the project architect. In the context of this thesis it also proves to be a very useful facility for the design of the non systemic parts of the building.

It is thus primarily a very efficient drafting tool, designed to operate optimally within a particular well defined environment. However it can also be considered as an aid to the design process since it provides the user with a very efficient tool for fast visualization of his idea without him having to waste time and energy in mechanical details related to the use of the system that abstract him from his design.

Some of the commands included in this facility contain expertise that is useful to the user when he performs certain simple design tasks and drafting routines within the previously described context. This expertise is embodied as a set of underlying rules which are accessible to the expert user and can therefore be reprogrammed according to the specific needs of the project he is currently working on.

4.2 Design Principles demonstrated by the facility

Besides the practical utility of this facility for system building as a tool that facilitates the design of the non systemic parts of the building, as well as its generic use within a general computing environment for architectural practice which make it a commonly used overlay to the graphics editor, this facility was developed in order to demonstrate two principles for the design of such tools. The first is related to a strategy for clearly identifying user requirements and most urgent user needs before starting to design the facility and thus not ending up turning it into a global tool and the second is related to the techniques with which expertise can, and should be built into such facilities.

The design of the tool proved that flexibility and efficiency (speed) are not incompatible for the design of such a facility. However, it is important to note that this was achieved because there was a very well specified set of performance specifications for the facility which derived from a similarly well defined scenario for its use. This is a tool which performs well in a very narrow and well defined field. If user needs expand, this implies the design of an other module that will be a more appropriate overlay to the graphics editor that will cover these new needs but will most likely not cover some of the area that the previous one did. It does not imply that the old overlay will expand to cover these needs.

This is a primary design consideration and is justified by the nature of the activity of designing that the architect performs. THe design process is very difficult to optimize, depending primarily upon the particular characteristics of each project as well as preferences and habits of the individual designer. Thus there is no limits in the amount of different subroutines one can program in order to perform specific tasks at various environments. However, this vast number of different subroutines that perform similar tasks, if they are all parts of a global graphics editor can only create confusion to the user and make the efficient use of the editor very difficult and improbable. Instead they should be contained in well defined packages-overlays. Each overlay should have a very narrow, clearly defined area of efficient performance. In this way the user simply selects the overlay according to his needs, and is automatically provided with those variations of the various routines for designing and drafting which are more suitable for the project he wishes to work on or the task he is currently engaged in.

The design strategy that the programmer follows in order to design such a module-overlay to the graphics editor is to clearly define a set of user requirements for his specific application, program those routines which are most urgently required in order to fulfill these user requirements and most likely to be generically utilized.

Thus this overlay, has a narrow and well specified purpose. Unlike others to be utilized for component building, does not aspire to automatically back the graphics editor with the maintenance of a data base which contains alphanumeric information. It is also inappropriate for designing with a grid and snap facility turned on. Since it does not aspire to be a general purpose enhancement of autocads graphics editor, expansions which would enable it to cover such areas and at the same time increase the size and the difficulty of use of the facility, were considered inappropriate and were ommitted.

The second principle that this facility stresses is the extent to which meta-commands which are programmed and run within the programming environment of the graphics editor can contain expertise and function as small and crude expert systems. The issue here is in what way the user should become aware of this expertise.

The idea is that yes, expertise can be incorporated efficiently in such commands and be of use to the architect. If one wants to make a modest statement ~~he~~ one may claim that these are simply small analysis routines that perform some calculation and test the result against a set of rules. According to the result of this test they provide the user with relevant information.

However these routines are more sophisticated than they at first seem and can be safely approximated to expert systems. Indeed they are very small and contain a very small, "minimal", knowledge base. Still several relatively trivial tests have to be performed during the design of the project which however do require a small amount of expertise. Furthermore they are repeated very frequently a fact which makes it very inconvenient to have to exit from the graphics editor and enter a expert system shell in order to be provided with an answer.

Expertise within a general computing environment for architectural practice cannot be isolated within smart modules of the system. Indeed there are areas where considerable amount of expertise must exist implying the creation of a large knowledge base to be used by an expert system, but there are more trivial tasks which still require expertise without justifying the need to built and have to access a full size expert system in order to get to it. It is only within the environment of the graphics editor that they this expertise can be found.

This is the kind of expertise that is appropriate to be embodied within a series of "meta-commands" which are invoked by the user when he wishes to perform a particular design operation. The evaluation routine is executed automatically at some stage of the execution of the "meta-command". It tests the users' input against a set of rules. The facility prompts the user only if necessary. Otherwise the user is unaware of its existence. This constitutes an different and quite innovative approach for dealing with several relatively trivial design tasks than attempting to access a conventional full sized expert system which has to be invoked explicitly by the user.

With the proposed approach, the system supports the user in his work by watching his moves and intervening only when some principle rules have been violated. It justifies his suggestions and very infrequently enforces them on the user. It incorporates knowledge stored as a set of rules, it justifies its answers and can be easily reprogrammed by the expert user. The expertise is also immediately accessible by the user, since the evaluation is performed automatically whenever the user performs geometric data manipulation in the process of designing a building.

This approach indicates a way to break up expertise and store it within the components of the overall computing environment. The other way that has still a lot of technical problems for its implementation, is the design and use of large expert systems that are invoked explicitly by the user to perform specific complicated tasks.

If this principle is followed and such "smart" overlays containing fragmented expertise embodied in several "meta commands" are developed, we shall move towards more intelligent computer environments. Also rapid developments in the area of expert systems and particularly in the now very limited capacity for input of knowledge and expertise into such systems will generalize the use of such systems as large intelligent components of a general computational environment for architectural practice. These systems, designed to perform efficiently a very specific task that requires a large amount of expertise will reside outside the graphics editor and will be appropriately interfaced with it. Thus expertise will be incorporated in the environment both within and outside the graphics editor. The parallel development of both areas will help create an intelligent computer environment suitable for architectural practice.

4.3 Description of the facility.

This application focuses on the manipulation of geometric entities. It consists of a set of generic functions that are linked together into complex "meta-commands" which perform these manipulations. It is entirely programmed in Autolisp which is a version of common LISP running within Autocad and including several specific commands suitable for the autocad environment as well as several limitations related primarily to memory allocation issues as well as difficulty in accessing Autocads data base.

The package includes facilities for the insertion of walls, for breaking of walls and creating openings, as well as inserting doors and windows in the openings. It manipulates entities at any angle and requires minimal input from the user.

The primary facility of the package inserts walls of any length and any width within the current layer of the drawing. The walls are specified by two point and can be of any angle. It also includes facility for the insertion of walls perpendicular to these walls and performs the connection automatically with the specification of only one point in the case of a "T" or "L" connection or two points in the case of the "+" connection.

The facility for breaking walls allows the user to create an opening of any size to an existing wall drawn at any angle. It incorporates features that allow the user to select very rapidly the starting and ending point of the opening without concern with accuracy (points can be anywhere on the plane at any distance from the wall) while the system snaps automatically to the point on the axis of the wall and performs accurately the manipulation of the drawing.

The facility for automating breaking of an existing wall and insertion of a door is based on a more complex subroutine. It contains the same sophisticated input facility for the breaking of the wall with the previous program but it also automatically inserts in the opening that is created, a door of a user specified type which has the length of the opening. The user is required to specify one more point to indicate the side of the wall where he wishes the door to open towards.

There is a similar facility that performs the same task with the previous one, however it also has expertise built into the program. When the user specifies the size of the opening for the wall to be broken and the door to be inserted, the program does not immediately perform the request but first goes through a sequence of calculations in order to determine the suitability of the opening for the insertion of a door according to a set of rules, "knowledge", that has been incorporated within it. If it finds that the width of the opening is well below an absolute minimum or well above an absolute maximum, making it impossible for a door to be inserted, it responds with a prompt that the size of the opening is too small or too large and kicks the user out of the subroutine without breaking the wall. If the opening is rather small or rather large but still possibly desirable in extreme situations it notifies the user of this fact and lets him or her decide whether this is intentional and he wants to continue, or not in which case instructions are provided that enable the user to get out of the subroutine. If the opening is of reasonable size, the program continues its execution without prompting the user and performs the rest of the subroutine which breaks the wall and inserts a door as already described.

This facility also introduces the concept of incorporating expertise within a program which displays information to the user whenever he violates some rule without him necessarily having to ask for it. This is very important since during a large number of simple design tasks there are rules which should not be violated and obviously the user cannot practically explicitly ask each time whether he has violated one. Furthermore in some cases it is much more efficient to detect whether some operation is undesirable before it is performed than after it has been completed.

In this facility, the expertise is available, incorporated into the program as a set of rules, in order to ensure that the doors inserted are of appropriate size for a person to walk through. However the existence of this expertise, only becomes apparent to the user when these rules are violated. The extent to which the expert system intervenes and the decisiveness with which it reacts, is based upon the extent of the violation. Also the system when making a particular action justifies its reaction to the user and furthermore, is sometimes ready to accept from him a deviation from the rule, i.e. treats him like an expert that must have some reason to deviate from the rule which is a legitimate approach to take in the case of design.

Thus the system acts as a decision support system supporting the user in his decisions and only very infrequently is forced to actually make a decision for him. Even in this case however, there is some alternative option for the user, as for example the use of the program which inserts doors without expertise, which for this reason remains an important component of the whole package. One should consider the inventiveness of a draftsman that may decide to utilize the subroutine which was designed for the creation of doors in an other way as for example for the creation of closets of cupboards which might be represented graphically by the same symbol. In this case the expertise embodied within the door subroutine would be considered a frustrating limitation.

However this infinite degree of flexibility cannot be the the goal in all applications. It would for example be considered a serious handicap in the case of structural analysis where the system has indicated failure of a component, as well as in several systems building applications where the rules for the interface of the components have to be respected at all cost,

The facility for insertion of doors in already existing openings of walls, operates primarily the same way as the previously described ones. It obviously does not have to break the wall, the user picks an opening instead of a wall and the program contains the expert facility. However it offers one more level of flexibility allowing the user to select the direction for the door to open independently from the side of the wall on which the door is inserted, making it an extremely handy tool for the insertion of doors in a floorplan with thick masonry walls.

The facility for insertion of windows in already existing openings as well as in walls that have to be first broken before the window is to be inserted, is structured the same way with the facility for the insertion of doors. However these are much simpler since there is symmetry over both the longitudinal and the lateral axis of the window when seen in plan and therefore the number of options that have to be checked each time are less.

In the case of windows, two sets of programs were developed. One inserts the window near the interior side of the wall and the other near the exterior side. All four programs for the insertion of windows, either for solid walls or for existing openings contain a facility for incorporating knowledge into them, which acts as some kind of empty expert shell and by default is not active or apparent to the user in any way. This shell contains some rules to which the user has to add some specific knowledge (most likely a few numbers) in order to have the system help him remain consistent with some rules in his design.

Thus, if the user decides that the size (width) of the windows of a particular building should not exceed a certain amount throughout his project, either due to environmental constraints (e.g. vernacular environment, Building regulations etc.) or for structural reasons (e.g. load bearing masonry walls) he or she can easily acquire a custom made facility suitable for the particular project. This capability of the system also indicates the advantages of working within a general computational environment for architectural practice which enables and facilitates in house programming, by utilizing a modular programmable graphics editor.

The menu is structured in two levels. The top level shows to the user the options available by this facility and also provides help on screen which explains both the use of the overlay as well as its structure. From the top-level menu the user is directed either to the walls menu or to the doors-windows menu.

The walls menu is used to invoke the set of "meta commands" that are utilized in the process of inserting (creating) walls, joining them as well as breaking them and creating openings. It also contains a help facility which appears on screen upon request and describes the function of each command. The doors and windows is used to invoke those commands which insert windows and doors either on existing walls or in existing openings. It also contains help facility for these commands similar to the one for the walls menu.

It is important to note that the help menus are designed in a way that utilizes both text and graphical representations in displaying information. It also makes use of the effect different colors have in attracting attention. The text is displayed in different colors according to the message displayed and two graphic figures accompany the description of what each command performs, indicating the geometric entity upon which the command performs the manipulation "before" and "after" the command is executed. Unfortunately colors cannot be reproduced in this document however figures of help menus are included in this document.

THE WALLS MODULE

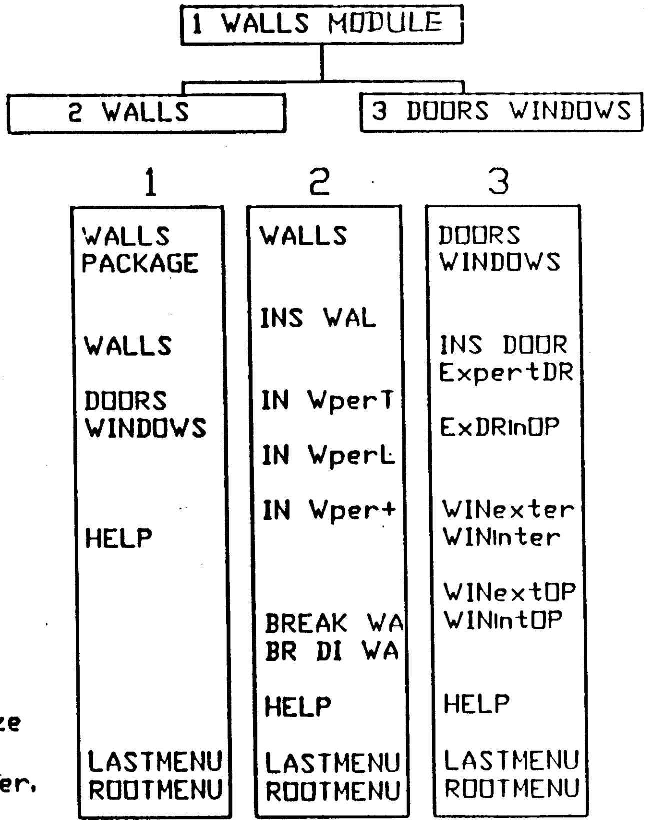
OVERLAY TO AUTOCAD'S DRAWING EDITOR. DEC 1985

This facility is usefull for fast and efficient creation of floorplans in projects where walls are likely to be placed at random angles or have various widths.

Such projects with only a small degree of uniformity are renovations or work in a vernacular environment

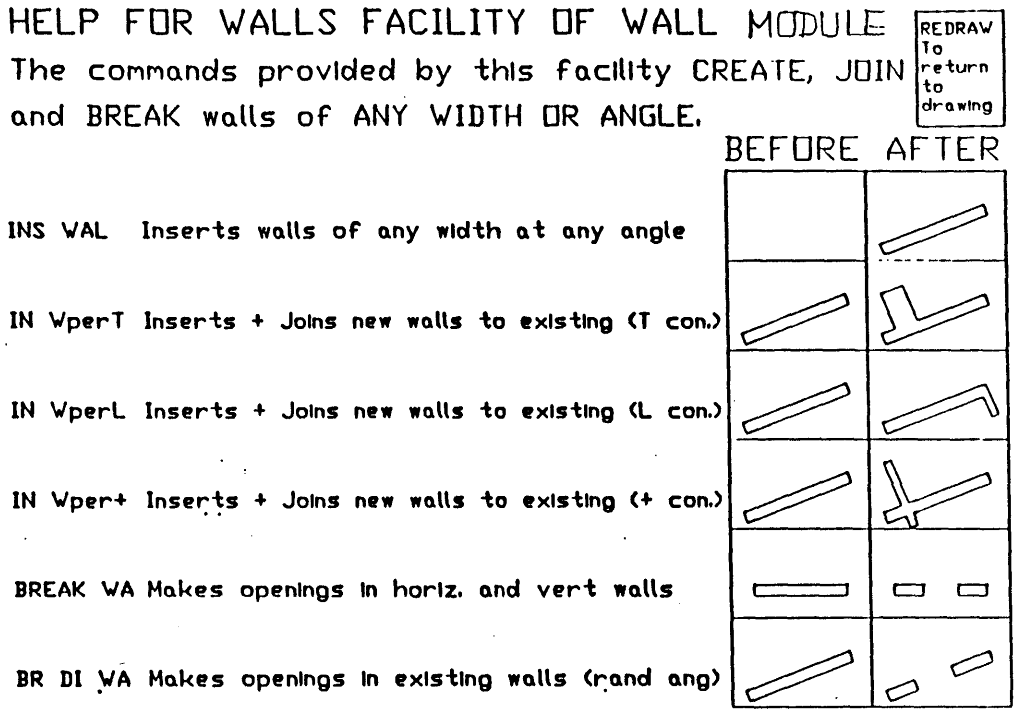
In the case of component building, this facility is usefull for the design of the non-systemic parts of the building.

The facility contains expertize The underlying rules are accessible to the expert user.



General structure of the Walls Module. Suitable for the design of non systemic building modules, or interventions on existing buildings.

(a)



HELP FOR WALLS FACILITY OF WALL MODULE

The commands provided by this facility CREATE, JOIN and BREAK walls of ANY WIDTH OR ANGLE.

REDRAW To return to drawing

BEFORE AFTER

INS WAL Inserts walls of any width at any angle

IN WperT Inserts + Joins new walls to existing (T con.)

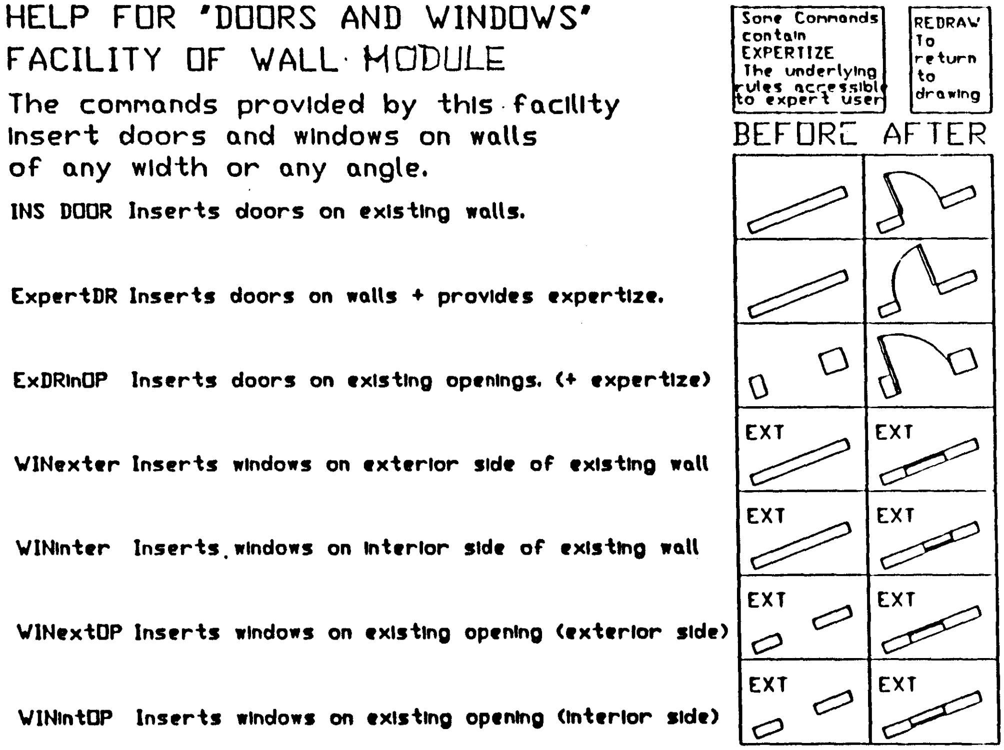
IN WperL Inserts + Joins new walls to existing (L con.)

IN Wper+ Inserts + Joins new walls to existing (+ con.)

BREAK WA Makes openings in horiz. and vert walls

BR DI WA Makes openings in existing walls (rand ang)

(b)



HELP FOR 'DOORS AND WINDOWS' FACILITY OF WALL MODULE

The commands provided by this facility insert doors and windows on walls of any width or any angle.

Some Commands contain EXPERTIZE The underlying rules accessible to expert user

REDRAW To return to drawing

BEFORE AFTER

INS DOOR Inserts doors on existing walls.

ExpertDR Inserts doors on walls + provides expertize.

ExDRInOP Inserts doors on existing openings. (+ expertize)

WINexter Inserts windows on exterior side of existing wall

WINinter Inserts windows on interior side of existing wall

WINextOP Inserts windows on existing opening (exterior side)

WINintOP Inserts windows on existing opening (interior side)

Help menus developed for the various levels of the Walls Module explaining the use of specific commands. (a) Walls facility. (b) Doors and Windows facility.

EPILOGUE

EPILOGUE

The generalization of the use of computers which we have recently started to experience, due primarily to a dramatic decrease in the cost of hardware, is likely to very rapidly change the image of the world we live in. It is hard to predict exactly what these changes might imply, and how each profession might be affected by them. Within this context, several individuals have attempted to envision future scenaria for the building process, the roles of the various professionals within this process and the role of the architect in particular.

This thesis did not aspire to make long term projections for the building process, issue which indeed involves a great degree of unpredictability. Instead, it has followed a down to earth approach, by observing the roles of the various participants within the currently prevailing building process and proposing a scenario that is likely to be realized in the near future.

This document was focused on a scenario for a computing environment that favors the application of Systems Building in Architecture. It has attempted to illustrate a way by which the cost of the design process can be significantly decreased, with a parallel increase in consistency of the output, and considerable savings of resources for the production of the built environment.

This innovative proposition is made possible by the integration of systems theory and powerful computer tools for the manipulation and handling of information. It is presented in the concept of a C.B.B.S. which assumes for its realization, the computerization of small architectural firms and the existence of an Open Computer System for Architectural Practice as presented in this document.

With the rapid developments in the area of Architectural Computing, Expert systems and Knowledge engineering, as well as the powerful hardware support for computer systems that is becoming more affordable, it is likely that we shall witness in the immediate future the necessary changes required for this scenario to be implemented.

Architectural firms are likely to be computerized in the near future for reasons not necessarily associated with the concepts presented in this thesis. Cheap microcomputer facilities will very rapidly invade architectural firms in order to help them eliminate waste of resources for the performance of repetitious jobs. A graphics editor is likely to be for still some time to come the central core of the computer system of an architectural firm due both to previous education of architects as well as the nature of their profession and specific interest in graphics and visual representation of the buildings they design.

It is very likely that the proposed scenario for an Open Computer System for Architectural Practice will be generically implemented, since it implies a particularly efficient use of computer facilities within an architectural firm. If this happens, the road will be open for the generic utilization of Systems Building in Architecture, by the use of Computer Based Building Systems.

All participants in the building process have increased interest in the development and use of C.B.B.S.'s. The product (systems) manufacturer has a good way to advertise and promote his product, facilitate its utilization, and develop a more efficient and easy to use building system. For him the C.B.B.S. is a much better marketable product than a paper based Building System. It is also more efficiently designed while the cost for its production remains the same or less of that of a conventional "Paper based" building system.

The project architect saves a lot of time and effort by utilizing a C.B.B.S. as opposed to a conventional B.S. since he is not forced to explicitly learn the rules of the building system he is going to use. He is also no longer responsible for ensuring that the design is consistent with the rules of the B.S. since the system performs this check for him. Furthermore the use of the C.B.B.S. implies the simplification of the design task, since a lot of design decisions of a minor scale are already made for him by the by the designer of the Building System, as well as the elimination of all repetitive tasks that are conventionally part of the responsibilities of an architectural firm, ie. production of construction documentation etc.

Therefore the cost of design services can be significantly reduced with the use of C.B.B.S. without any reduction in the quality of the output. This allows the architect to be able to provide their services for the design of a much larger portion of the built environment. This not only is important for the architectural profession which will thus be able to find itself able to play a more decisive role in the development of the built environment, but is of common interest to all participants in the building process and especially to the users.

The users will find that their needs will be more efficiently met, since the reduced cost of the design process will allow them to hire -directly or indirectly- professional architects whose services they could not previously afford, in order to assist them in the creation of a more human built environment. Furthermore they will find that the cost of the buildings will be significantly decreased while their efficiency will be increased due to utilization of efficient and previously designed building systems.

Systems Building, if efficiently applied in the building process in a way that allows for an open system approach, can produce an output which will not only conserve valuable resources for its creation and maintenance, but will also prove very satisfying to the user. Associations with a mass-produced and sterile environment will no longer be relevant since the utilization of small and cheap computer facilities will enable the utilization of the systems approach to building for relatively small projects too. The application of systems building in Architecture will not depend upon the existence of conditions that are only likely to be found in rare occasions, i.e. large scale of project or need for speed etc.

The two notions presented in this thesis, that of the Computer Based Building System and that of the Open Computer System for Architectural Practice, constitute an appropriate environment for the application of Systems Building in Architecture. It is hoped that this work will serve as a starting point for relevant future research in this promising area.

GENERAL BIBLIOGRAPHY

GENERAL BIBLIOGRAPHY

1. Abelson, Harold, Sussman, Gerald J. "Structure and Interpretation of Computer Programs" The MIT Press, Cambridge Mass. and London, England.

2. ACADIA Conference Proceedings. (Spring 1985).

3. Ackoff, Russel L. "Redesigning the future." John Wiley and sons.

4. Addis, T.R. (1981) "Definition of an Expert System" BCS Specialist Group on Expert Systems Newsletter No 3 Spring/Summer 1981.

5. A.J. (1983) "Micro-computing for the architect" Architect's Journal 19.1.83

6. Alexander, C. (1968) Systems Generating Systems." Architectural Design.

7. Althuser, L. "Positions (1964-1975)." Editions Sociales, Paris 1976.

8. Archer, L.B. (1981) "A view of the nature of design research" in : Jacques R. & Powell J. (eds) "Design : science : method : proceedings of the 1980 DRS conference" Westbury House, Survey.

9. Berger, S.R. (1980) "Artificial Intelligence and its impact on computer-aided design" Design studies 1 3 p 166-171.

10. Bertalanfy, L. von. (1980) "General System Theory." George Braziller New York.

11. Bijl, A. (1978) "Machine discipline a& design: notes on experience

of CAAD systems". CAD' 78 Proceedings, I.P.S. Science & Technology Press, Guildford.

12. Bijl, A., Stone David, Rosenthal David, (March 1979) "Integrated CAAD Systems" Final Report of DoE funded research project EdCAAD studies.

13. Bijl, A (1980) "The Revolution is here to stay" C.A.D. 12 3 p.107-114.

14. Bijl, A. (1982) "Dumb drawings and knowledge engineering" CAD' 82 Proceedings, I.P.C. Science & Technology Press,Guildford.

15. Braid, I.C. (1975) "Six Systems for Shape Description and Representation - A Review" University of Cambridge CAD Group, Document 87, May 1975.

16. Brandle, K. (1974-1976) "The Systems Approach to Building, a case of information." Associated schools of Architecture Learning Package.

17. Bridges, A. (1982) "CAAD in education & Practice". CAAD' 82 Proceedings, I.P.C. Science & Technology press, Guilford.

18. Buchaman, D.A.& Boddy D. (1983) "Organisations in the Computer age: Technological imperitives & strategic choice" Gower Press, Aldershot.

19. Buchanan, B.G., and Feigenbaum, E.A. (1978), "DENDRAL and META-DENDRAL : Their Applications Dimension", Artificial Intelligence, vol.11.

20. Camous, R. "Information about building Systems." IF 2292.

21. Campion, D. (1980) "A computer-aided draughting/scheduling system in a medium sized architectural practice". CAD' 80 Proceedings, I.P.C. Science & Technology Press, Guildford.

22. Chalmers, J. (1972) "The Development of CEDAR" Proceedings of International Conference on Computers in Architecture, York, 1972.

23. Chalmers, J., Sampson P. Webster G. (1974) "Data Structure Used In CEDAR" in Programming Techniques in CAD" Sabin. M. (ed) National Computer Centre.

24. Charlesworth, D., Webster G., (1976) "CEDAR 3: Basic software for Computer Aided Design" CAD 76 Proceedings.

25. Churchman, C. West. (1968) "The Systems Approach." New York: Dell.

26. Collins, E.B. (1980) "Experience in the use of the B.D.S. system (OXSYS) on a major project in a multidisciplinary consultancy". CAD' 80 Proceedings, I.P.C. Science & Technology Press, Guildford.

27. Cronberg, T. (1975) "Performance requirements for buildings- a study based on user activities." Sweedish Building Research Summaries.

28. Cross, N. (1977) "The Automated Architect" Pion Limited. London.

29. Daniel, P.T. (1973) "A Plant and Buildings Draughting and Scheduling System," in J Vlietstra and R.F. Wielinga (eds), Computer Aided Design. Amsterdam, Holland : Norht Holland Publishing Co.

30. D' Arcy, Richard (1979) "The Oxford System and its Application." Proceedings of the International Conference on Computers in Architecture.

31. Davison, J.A. (1980) "The tactical & Strategic impact of RUCAPS implementation in the design office". CAD' 80 Proceedings, I.P.C. Science & Technology Press, Guildford.

32. Dietz, A.G.H. (1971) "Industrialized Building Systems for Housing." M.I.T. Press.

33. Duda, R.O.,Gaschnig,J., and Hart, P.(1979),"Model Design in the Prospector Consultant System for Mineral Exploration", Expert Systems and AI Applications.

34. Eastman, Charles, M. (November 1972) "Requirements for Men-Machine Collaboration in Design", Carnegie Mellon University, Rescarch Report No.35.

35. Eastman, C. et al (1974) "An Outline of the Building Description System" Inst. for Physical Planning, Carnegie-Mellon University.

36. Eastman, C. (1975) "Preliminary Documentation for BDS-10" Inst. for Physical Planning, Carnegie-Mellon University.

37. Eastman, C. (1975) "General Purpose Building Description Systems" Inst. for Physical Planning, Carnegie-Mellon University Research Report 57, Sept 75.

38. Ehrenkrantz, E.D. (May 1970) "The Systems to Systems." AIA Journal.

39. Ehrenkrantz, E.D. "Building Systems Development." Bulletin.

40. Emery, F.E. (1978) "Systems Thinking." Penguin Books.

41. Feigenbaum, E.A. (1977) "The Art of Artificial Intelligence,

Themes and Case Studies of Knowledge Engineering", Fifth International Joint Conference on Artificial Intelligence.

42. Feigenbaum, E.A. (1981) "Expert Systems in the 1980's" in A.Bond's "Machine Intelligence"

43. Ferguson, F. (1975) "Architecture, Cities and the Systems Approach", George Braziller, New York.

44. Friedman, Y. (1971) "Pour l' Architecture Scientifique." Pierre Belford Art-Action-Architecture, Paris.

45. Gero, John S. (1977) "Computer Applications in Architecture", Applied Science Publishers, London.

46. Gero, John S., (1983) Akiner Tuncer V., Radford Anthony D., "What's what and what's where: Knowledge Engineering in the representation of buildings by Computer." PARC 83 Proceedings.

47. Greig, Jonathan (1983) "possibilities of Low Cost Computer Systems." PARC 83 Proceedings.

48. Hamilton, I. & Burdett J.R.F. (1982) "Computer draughting system in construction __ the byers' problem. CAD' 82 Proceedings, I.P.C. Science & Technology Press, Guildford.

49. Harper, D.R. (1978) "Building: The Process and the Product." The Construction Press.

50. Hoskins, E.M. (1972) "OXSYS : An Integrated Computer Aided Building System for Oxford Method. "Proceedings of the International Conference on Computers in Architecture. London : British Computer Society.

51. Hoskins, E.M. (1973) "Computer aids for System Building" Industrialization Forum, Vol.4, no.5

52. Hoskins, E. "The OXSYS System" (1977) in Gero J. (ed.) "Computer Applications in Architecture" Applied Science Pub. 1977.

53. Hoskins, E.M. and Bott, M.F. (1972), "DHHS :Development Planning System and Production Documentaion System "Proceedings of the International Conference on Computers in Architecture. London :British Computer Society.

54. Jones, E.M.(1982) " The design of a complex building using an integrated CAAD system". CAAD' 82 Proceedings, I.P.C. Science & Technology Press, Guildford.

55. Kay, N.M.(1983) "Innovation, markets and hierarchy". Proceedings of Lovy Waves and International Development Conference, Edinburgh, Development Studies Association.

56. Klam, A. (1982) "Space and Material: Towards an Architectural Typology." Thesis, M.I.T.

57. Kociolek, A., Radwanski A., (1979), "Computer-Aided design in industrial residential construction" Proceedings of the International Conference on Computers in Architecture.

58. Lansdown, J. (1982b) "Expert systems: their impact on the construction industry". R.I.B.A. Conference Fund, London.

59. Lazlo, E. (1972) "Introduction to Systems Philosophy; Toward a New Paradigm of Contemporary Thought." Harper Torchbooks, New York, Evanston, San Francisco, London.

60. Little, S.E. (1982) "Architectural Design process in the 1980's : transition and the need for re-appraisal". Design Policy Conference, R.C.A. London.

61. Little, S.E. (1984) "The Organisational implications of CAAD" CAD'84 Proceedings, I.P.C. Science & Technology Press, Guildford.

62. Llewelyn, A. (1982) "CAD'82 Proceedings,I.P.C. Science & Technology Press, Guildford.

63. Martin, B. (1969) "Standards and building. " RIBA Publications Limited.

64. Meager, M.A. (1972) "Computer Aids in Hospital Building" Proceedings of the International Conference on Computers in Architecture. London : British Computer society.

65. Meager, M.A. (1973) "The Application of Computer Aids to Hospital Building," in J.Vlietstra and R.F.Wielinga (eds.) Computer-Aided Design. Amsterdam, Holland: North-Holland Publishing Co.

66. Mitchel, William J. (1977) "Computer Aided Architectural Design" Van Nostrand Reinhold Company.

67. Mumford, E. & Wier (1979) "Computer systems in work design : the ETHICS method." Associated Bussiness Press, London.

68. Nash, J.H. (1982) "Graphic interactions with database systems" CAD'82 Proceedings, I.P.C. Science & Technology press, Guildford.

69. Patterson, J.W. (1972) "Aids to improve the Designer's Control of Cost and Performance of Buildings, and to speed the Pre-Contract and Post-Contract Processes," in W.J.Mitchell (ed.) Environmental Design:

Research and Practice, Proceedings of the EDRA3/AR8 Conference. Los Angeles, California : School of Architecture and Urban Planning, UCLA.

70. Patterson, J.W.(1974) "An Integrated CAD System for an Architect's Department." Computer Aided Design, Vol.6 no. 1 (Jan. 1974).

71. Pollalis, Spiro N., Habib W.S. (1985) "Selection and Predimensioning of Structural Systems". Civil Engineering Department, Norhteastern University.

72. Popple, H.E., (1977), "The Formation of Composite Hypothesis in Diagnostic Problem Solving", Fifth International Conference on Artificial Intelligence.

73. Purcell, Patrick A. (1975) "Computer Aided Architecture in the United Kingdom" in N.Negroponte (ed.), Computer Aids to Design and Architecture. New York : Petrocelli/Charter.

74. Purcell, Patrick A. (1980) "Computer education in Architecture" C.A.D. 125 p.239-251.

75. Purcell, Patrick A. (1983) "The Data Station: An Integrated graphic Information facility." PARC 1983 Proceedings.

76. Ray-Jones, A. (1968) "Computer Development in West Sussex." Architects' Journal 12 February 1968.

77. Richens, P (1974) "OXSYS: Computer-Aided Building for Oxford Method." Cambridge, England: Applied Research of Cambridge.

78. Rosenthal, David S.H. (1977) "The PIM System - Integrated Display facilities" EdCAAD Rept. 1977.

79. Rosenthal, David, S.H. (1979) "Building Description techniques for rationalized traditional Construction". Proceedings of the International Conference of Computers in Architecture.

80. Ruffle, Simon (1983) "A better man-machine interface to replace the keyboard." PARC 1983 Proceedings.

81. Russel, B. (1981) "Building Systems, Industrialization and Architecture." John Wiley and Sons.

82. Schmid, T. and Testa C. (1969) "Systems Building" Les Editions d'Architecture, Artemis, Zurich.

83. Shortliffe, E.H. (1976), Computer-Based Medical Consultations : MYCIN, American Elsevier Publishing Co., New York.

84. Simon, A.H. (1982), The Science of the Artificial, MIT Press, Second Edition, Cambridge Massachusetts.

85. Spillenkothen, R. and Rennea, J. "Standardization and Industrialization." IF 2133.

86. Swinson, P.G. (1982) "Logic programming : a computing tool for the architect of the future." CAD 14 2 p.97-104.

87. Techniques at Architecture. Vol.327,328. Industrialization Ouverte."

88. Terner, I. and Turner, J.F.C. (1972)"Industrialized Housing." Department of Housing and Development office of International Affairs.

89. Tzannetakis, Charilaos P. (1983) "Building Systems in Architecture" Masters thesis.

90. Vien, Therese (1985) "Investigation of an Integrated System for Architectural Design" Masters Thesis.

91. Walters, R.J (1982) "Architect as data-base manager; some user-experience of an integrated CAD system".

92. Walters, R.J. (1983) "Towards an End user view of Design Systems" Park 83 Proceedings.

93. Webster, R. (1983) "Expert Systems" Personal Computer World 6 1 p.118-119.

94. Winston, P.H. (1977), Artificial Intelligence, Addison - Wesley Publishing Company, Reading, Massachusetts.