CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

THE CADDIE PROJECT:
Applying Knowledge-Based Paradigms To Architectural Layout Generation

by

Paul Scott Chinowsky

TECHNICAL REPORT
Number 54

May, 1991

Stanford University

THE CAADIE PROJECT:

APPLYING KNOWLEDGE-BASED PARADIGMS TO ARCHITECTURAL LAYOUT GENERATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF CIVIL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By

Paul Scott Chinowsky

May 1991

ABSTRACT

The development of systems which produce architectural layouts from a set of spaces has been an objective of design research efforts for more than two decades. These research efforts have spanned a diversity of computational approaches - from the use of mathematical optimization techniques, to the present emphasis on knowledge-based paradigms. The evolution to these knowledge-based paradigms has brought about the initiation of numerous issues to all aspects of layout generation research. Considerations related to one segment of this domain, the conceptual design of architectural floorplans, are addressed in the Computer-Aided Architectural Design Expert (CAADIE) project.

The CAADIE project endeavors to explore the potential for utilizing knowledge-based paradigms in the development of conceptual design diagrams. The system is being developed as a cooperative design tool with which designers can obtain assistance during the creation of space relationship diagrams. The system complements the expertise of designers by using knowledge provided by the designers, in conjunction with the CAADIE knowledge base, to address the multiple design issues and iterative design process associated with the layout generation task. The CAADIE project incorporates several objectives including the development of a cooperative design environment, the development of a model to represent the knowledge and information used by designers during layout conceptualization, and the implementation of a control structure which facilitates a cooperative control strategy. Based on these objectives, a prototype of the CAADIE system has been developed. The prototype demonstrates the viability of the CAADIE concept and the potential for applying knowledge-based paradigms in the earliest phases of architectural design.

ACKNOWLEDGEMENTS

The completion of a doctoral dissertation requires a significant commitment. However, the commitment to pursue this research is not made exclusively by the student. Rather, the completion of a doctoral dissertation represents a combination of dedication from the student, guidance from the committee members, and support from family and friends. Without each of these components, the student will be unable to devote the time and energy required to complete this major undertaking. However, during the course of the research, there are never enough opportunities to adequately thank those people who have provided the support and guidance to complete the work. Rather than complaining when the student devotes ever increasing amounts of time to completing the work, the people around the student provide the additional levels of support required to assist the student. It is to these people that I direct the following acknowledgements, and to whom I say thank you for providing the support and guidance I needed to achieve this accomplishment.

First, to my committee members from the Civil Engineering Department, Kincho Law, Ray Levitt, and Paul Teicholz, thank you for keeping me on track and providing the guidance I needed to bring CAADIE from an idea to a tangible result. To the fourth member of my committee, Jens Pohl, thank you for having the patience over the last seven years to continue to show me that there was more that I could accomplish. I would never have been writing this without your encouragement.

Second, a thank you to my friends who have both provided invaluable input to this research, and made life at Stanford an unforgettable experience. Thank you to Geoff Cohen, Hossam El-Bibany, Martin Fischer, Karen Hansen, Kelly Fergusson, Ana Garcia, and Leyla Unerdam. Also, a special thank you to Chris Madison for making everyday at CIFE a little more fun. Finally, to Art Chapman and John

Harrigan, I hope the friendship we have formed over the last nine years will continue for many years to come.

Third, to my family, I don't think there are enough things I could say to adequately express my thanks for all of your support and encouragement (even when you didn't understand what I was doing).

Finally, three people deserve very special thanks. To Thomas and Deb, don't ever forget that whatever you do I will be thinking about you, and "Go for it baby with a jump shift bid!!!". And to Melissa, let me just say thank you, I love you, and I hope I can give you the same kind of support you have given me.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

PROJECT INTRODUCTION

The development of systems to produce architectural layouts from a set of spaces has been a design research objective for over two decades. Researchers have recognized the potential impact of integrating computer-aided design tools into the layout generation process including an increase in the number of alternatives considered, and a decrease in the number of omitted design requirements. However, previous research efforts have failed to yield a system which addresses the spectrum of issues associated with the layout generation process. Consequently, the full potential of layout generation systems has yet to be demonstrated or tested. The Computer-Aided Architectural Design Expert (CAADIE) research effort proposes to diverge from this continuum by addressing the layout generation problem through the application of knowledge-based representation and reasoning paradigms.

The CAADIE project endeavors to explore the potential for utilizing knowledge-based representation and reasoning paradigms to model and capture the knowledge and information associated with generating architectural layouts. This approach combines artificial intelligence advances with traditional architectural design theory to create configurations portraying established design principles and concepts. The resulting prototype system will work as a computer-aided design tool, in cooperation with designers, to address conceptual design issues. Specifically, the prototype will facilitate the generation of space relationship diagrams (i.e., bubble diagrams). It is intended that CAADIE will complement designer expertise by using experiential knowledge provided by the designer, in conjunction with the CAADIE knowledge base, to address the multiple design issues and iterative design processes associated with the layout generation task.

Project Objectives

The overall goal of applying knowledge-based paradigms to architectural layout generation incorporates several project objectives. The following sections introduce the motivations behind each objective.

- **Combine the reasoning strengths of designers with the information processing strengths of computers**

Designers and computers each bring a series of strengths and weaknesses to the layout generation problem. Computers contain an almost unlimited capability to process information within structured guidelines. For example, given a problem and a set of specific parameters, computers can solve the problem in accordance with all documented constraints. However, current computer implementations contain no inherent reasoning capabilities to allow them to significantly diverge from predetermined problem solving routines. In contrast, designers adapt to changing problem circumstances through analytic reasoning and the ability to effectively organize design information. Conversely, designers are restricted in their cognitive ability to address more than a limited number of problem parameters at any given time (Simon 1981).

Based on these relative strengths and weaknesses, the focus becomes directed towards the possibility of capturing both strengths in a knowledge-based layout generation system. Capturing these strengths through knowledge-based paradigms will provide a basis for complementing the respective reasoning and information processing capabilities, while surmounting stated reasoning and problem solving weaknesses. Investigating the viability of this approach is a fundamental objective of the CAADIE research effort.

- **Create a participatory design environment**

Layout generation research should not automatically imply that designers will be excluded from design process and system control issues. Designers bring a significant amount of experience and intuition to each layout generation problem. Excluding the possibility for designers to impart this experience and intuition during the layout generation process precludes designers from adapting the system to individual design preferences. Additionally, the exclusion of this participation precludes the possibility of a system adapting and expanding its knowledge base through automated learning procedures. Finally, this exclusion alters the system's role from an interactive design assistant to an independent design generator. Therefore, a primary CAADIE objective is the creation of a participatory design environment to encourage designer participation through flexibility in control, generation, and knowledge adaptation facilities.

- **Identify and classify the areas of information and knowledge used during layout generation**

The identification of layout information and knowledge will provide a basis for modelling the layout generation process. Specifically, the information areas impacting layout generation problems, and the knowledge used to transform this information into layout configurations, represent the essential components required to generate layout alternatives. Emerging from these individual areas will be a viewpoint characterizing the way designers reason about the layout generation process. This viewpoint may then be contrasted with existing layout generation models to determine the validity of approaching the problem from a reasoning basis rather than a computational basis.

- **Create a framework to incorporate layout information and knowledge within a knowledge-based system**

A knowledge-based layout generation system requires the information and knowledge impacting configuration development to be captured within appropriate representation paradigms. The CAADIE project endeavors to support this requirement through the development of a knowledge model. This model will permit the system to access established design knowledge and typical layout parameters. This information and knowledge will then provide the basis for design assistance. Additionally, the framework will provide a starting point from which to examine the applicability of this representation paradigm to related design disciplines.

- **Develop a flexible control structure to facilitate cooperative layout generation**

A design team comprises experts from various domains. These experts contribute specialized knowledge to address particular design issues. Through this collaboration, the team ensures that sufficient expertise exists to derive design solutions. The final project objective addresses this collaborative problem solving style by emphasizing the development of a cooperative control architecture. This structure will provide an underlying framework for coordinating input from multiple knowledge-based design components, as well as, the flexibility to increase or decrease the number of components involved in the layout generation process. Thus, the cooperative nature and flexible structure evident in design teams will be reflected within the CAADIE system.

Overall Project Scope

Several overall project boundaries have been established to limit the scope of the project. These restrictions do not reduce the project contributions. Rather, the restrictions facilitate the development of a proof of concept system which validates the research concepts within a realistic, but simplified, context.

- **Address a single conceptual design stage**

The design of a building incorporates multiple design stages which combine to form overall design phases (figure 1-1). Each of these stages incorporates a

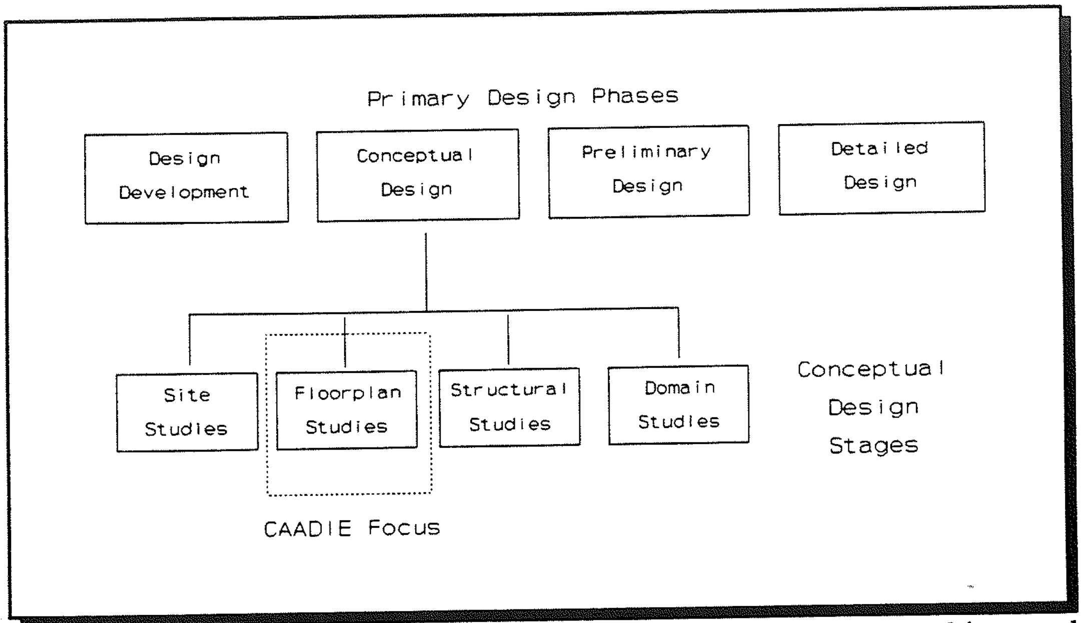


Figure 1-1: CAADIE focuses upon the conceptual design of architectural floorplans

unique set of issues and design requirements. To ensure that CAADIE effectively addresses these issues and requirements, the project is restricted to a single design stage, the conceptual design of architectural floorplans.

- **Address the layout information requirements of a single building type**

Particular building types contain sets of spaces which typically exist in each instance of that building type. To restrict the number of space types defined in the knowledge base, the system restricts its view to a single building type, university research buildings (i.e., university buildings used for teaching and general research purposes).

- **Limit the design domains captured within the knowledge base**

The development of a layout generation methodology and a cooperative problem solving theory are addressed within this research study in terms of complete system implementations. However, in the CAADIE prototype, the number of domains is limited to space planning, acoustics, security, privacy, and daylighting, to reduce the knowledge acquisition requirements.

- **Develop layouts in a two-dimensional environment**

The configurations generated in the CAADIE system are restricted to a two-dimensional basis. This restriction reduces the reasoning requirements for, and emphasis on, three-dimensional form considerations such as building step-backs, room heights, and shadows.

- **Address single-storey layout generation problems**

The generation of multi-storey layouts requires an examination of both individual floor requirements and above-below relationships. To reduce the spatial reasoning requirements within the system, configurations are restricted to single-storey considerations. This restriction does not preclude the system from demonstrating layout generation capabilities associated with the conceptual design stage of floorplan development.

Research Methodology

The CAADIE research effort achieves the previously described objectives through the six primary stages illustrated in figure 1-2.

- **Definition of Objectives**

The initial methodology stage sets the overall project guidelines. The project goals, project objectives, and problem definition provide the foundation for the CAADIE research effort.

- **Definition of Scope**

The scope definition stage addresses the overall project and specific prototype limitations. This area incorporates the restrictions related to the project context including the domain selection. The latter specifies prototype restrictions including

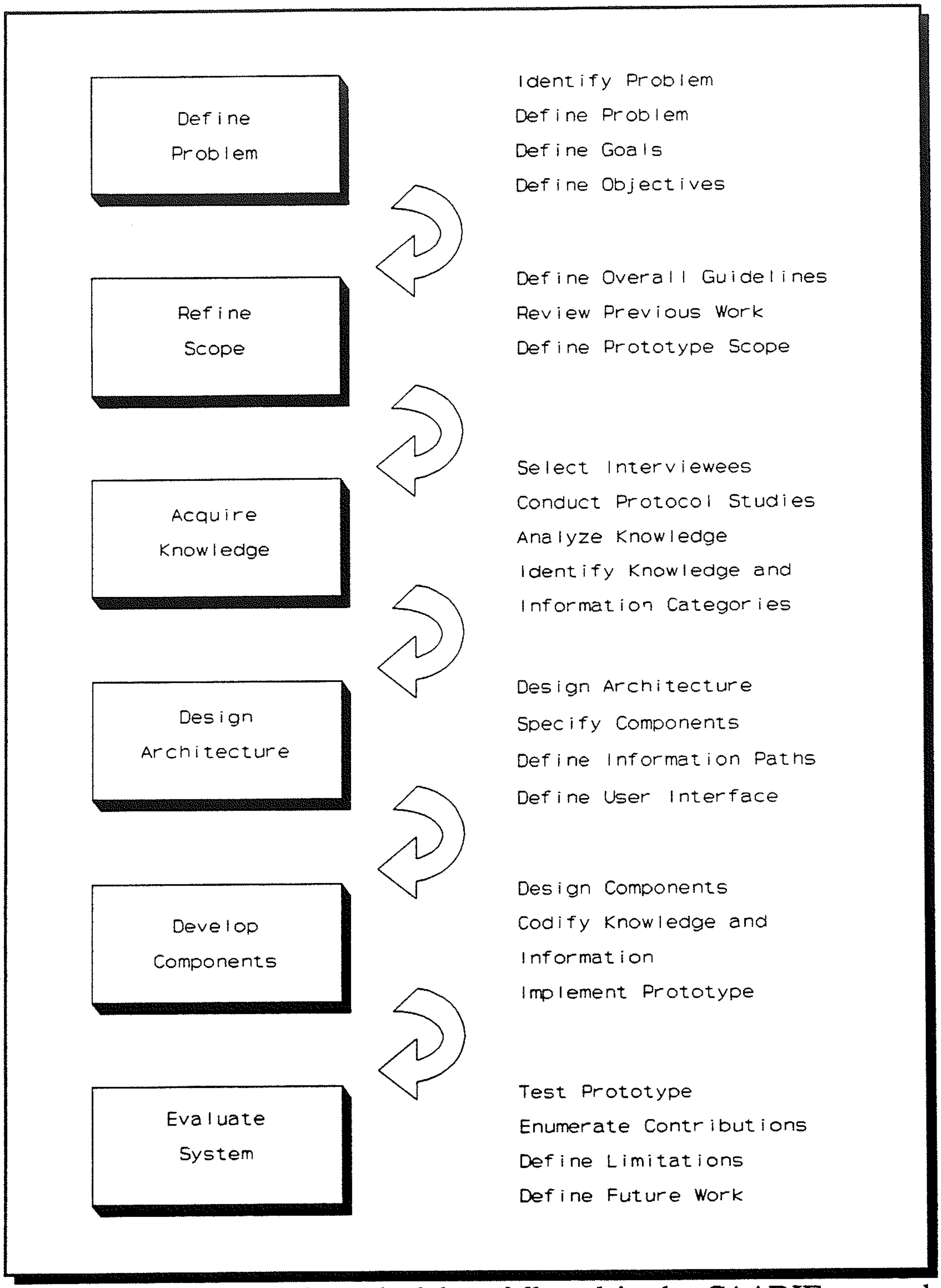


Figure 1-2: The research methodology followed in the CAADIE research effort including major sections and individual components

the definition of specific design attributes and layout generation stages which will be addressed within the layout generation process.

- **Knowledge Acquisition**

Following the project definition phase, the focus shifts to the acquisition of domain knowledge and information. Within this phase, interviews with designers are conducted to analyze the layout generation process, and obtain an understanding of areas where designers require assistance.

- **System Design**

The system design phase identifies the individual components required for a participatory design environment, together with the associated information flows. Additionally, the user interface requirements are analyzed to provide implementation guidelines.

- **Component Implementation**

The component implementation phase encompasses prototype design and implementation tasks. The objective is to implement a working prototype to illustrate and validate the CAADIE research concepts. Specific project decisions made within this stage include the selection of hardware and software platforms and the determination of graphic representation requirements.

- **Project and Prototype Evaluation**

The final project phase evaluates the overall research contributions and the CAADIE prototype based on the initial project objectives.

Reader's Guide

The dissertation progresses from a general problem overview, to an introduction of the CAADIE system, and finally, to specific component descriptions. Chapter 2 introduces the layout generation problem, and enumerates the specific design and computational issues impacting layout generation system development.

Chapter 3 provides a transition to the CAADIE approach to layout generation by reviewing and analyzing previous layout generation research efforts according to the issues defined in Chapter 2. The research review then contrasts the previous approaches with the principal CAADIE research objectives. Chapter 4 describes the knowledge acquisition process conducted in this project, and introduces the system architecture developed for the prototype.

The dissertation emphasis narrows at this point to focus on prototype component descriptions. Chapter 5 introduces the knowledge and information categories found in the knowledge model, together with a discussion of the knowledge model implementation. Chapter 6 provides an overview of the layout generation process including descriptions of the analysis, synthesis, and evaluation stages. Chapter 7 introduces the CAADIE user interface. This chapter enumerates the issues associated with developing user interfaces for layout generation systems, and illustrates implementation examples from the CAADIE user interface. Finally, Chapter 8 concludes the dissertation with a discussion of research contributions, prototype limitations, and opportunities for future research.

CHAPTER 2

THE LAYOUT GENERATION PROBLEM

Problem Definitions

The fundamental objective of creating a definition for the layout generation task is to separate this task from the overall architectural design problem. This separation is important due to the diversity of activities which encompass the architectural design process. Addressing one task within this process such as layout generation, requires a definition which isolates individual characteristics, issues, and goals. Representative layout generation problem definitions designed to accomplish this objective have been given as follows:

> Space-planning is that aspect of environmental design which is concerned with the physical arrangement of objects and spaces within a room, building or site to fulfill the requirements of diverse human activities (Henrion 1978, 175).
>
> ... a space-filling location problem is defined as a problem which has the goal of the placement of a set of subspaces in a particular larger space, subject both to a class of location requirements and to the constraint that the subspaces must entirely fill the larger space (Grason 1970a, 175)[1].
>
> Given a data structure capable of representing a range of building designs find a state of the data structure (i.e., a particular design

---

[1] In this definition, Grason is restricting his view of the problem to a class of problems which require the spaces in a given configuration to fill the entire building space given.

> solution) such that specified objectives and/or constraints are complied with (Mitchell 1977, 425)[2].
>
> Given a list of rooms, dimensional constraints on their areas and extensions ..., and topological constraints on interconnections of rooms or their access to open air, find: one or more floor plan(s) composed of the rooms so as to satisfy the constraints (Galle 1986, 21).
>
> Given:
> n spaces to be 'densely packed' within a given area: that is, allocated such that no two spaces overlap and no part of the given area remains which does not belong to one of the given spaces;
> 'dimensional constraints' restricting the form or dimensions of the given spaces: ... and 'topological constraints' requiring direct adjacencies between pairs of spaces or between a space and the outside:...
> find: spatial arrangements which satisfy the given constraints (Flemming 1978, 215).

Common Definition Characteristics

Two common characteristics emerge from the previous definitions: requirements emanate from the spaces, and all specified requirements must be addressed within the layout generation process. The first characteristic defines the scope of the layout generation problem. Individual requirements such as adjacencies, security, and acoustics, permit the problem to be viewed in terms of requirement compliance. Specifically, the addition of spaces to a potential configuration focuses upon generating placement options which comply with the given design requirements. Viewed at this level of granularity, the success or failure of a configuration is determined exclusively by its compliance with the aggregated layout requirements. The advantage of this approach is that requirements related to design decisions such as floor and wall finishes, and overall project decisions such as economic viability,

---

[2] In this definition, Mitchell is emphasizing the viewpoint of creating automated layout generation systems, rather than emphasizing a general architectural design viewpoint.

may be deferred until later in the design process. Subsequently, the layout generation task is isolated to the development of configurations which comply with space-specific requirements.

The second characteristic emphasizes guidelines for executing the layout generation process. Specifically, these guidelines ensure that all requirements associated with individual spaces are addressed during the determination of placement option compliance. Thus, the resulting solution will either comply with all stated requirements, or include trade-offs where compliance with multiple constraints is not feasible due to requirement conflicts. In contrast, a process addressing only a segment of these requirements such as adjacencies, may achieve compliance with the stated adjacency requirements, however, the solution will be unacceptable to a designer due to its limited focus and disregard for the remaining design program.

The combination of these two emphasis areas has provided a basis for many research efforts in the layout generation domain. However, the ambiguous nature of the definitions in terms of addressing specific layout requirements, has often led to researchers simplifying the problem by altering the definitions to explicitly limit the number of requirements addressed in the problem. This form of modification reduces the scope of the problem to isolate specific research objectives. Subsequently, the layout generation systems developed within this limited scope focus on a limited segment of the issues arising in the layout generation process. For example, in the initial drive to develop layout generation systems, one problem characterization narrowed the definition to focus on the synthesis of configurations based entirely on adjacency constraints (Armour and Buffa 1963). Through this narrow definition, solutions minimized the distances people had to travel in their typical visits to various spaces within the building. However, the remaining design influences were eliminated from the layout generation process.

The fundamental flaw with this approach to defining the layout generation problem, and similarly, to developing layout generation systems, is the limited view obtained of the problem. Layout generation cannot be viewed in the context of a single design constraint such as adjacencies, or in the context of a single design phase such as layout synthesis. Rather, the task must be viewed in terms of the complete spectrum of issues which impact layout generation. The following section represents one answer to achieving this viewpoint by explicitly defining the issues which layout generation systems are required to address if they are to succeed in addressing the true complexity of the layout generation problem.

Layout Generation Issues

Based on automated layout generation research conducted over the last three decades, existing architectural design theory, and architectural design process studies conducted by the author, two types of fundamental layout issues emerge: design issues and computational issues. These issues define the primary considerations which must be addressed in layout generation research efforts. The degree to which these considerations are effectively addressed, defines the scope, capabilities, and characteristics of layout generation systems.

Design Issues

Design issues comprise both design influences and design processes. Design influences encompass the multiple attributes, external influences, design knowledge, and design concepts which combine to create a unique set of requirements, or design program, for each layout situation. Similarly, design process issues encompass the multiple stages and design phases employed by a designer during layout evolution. The following descriptions introduce these issues together with a brief discussion of the impact these issues have on the development of layout generation systems.

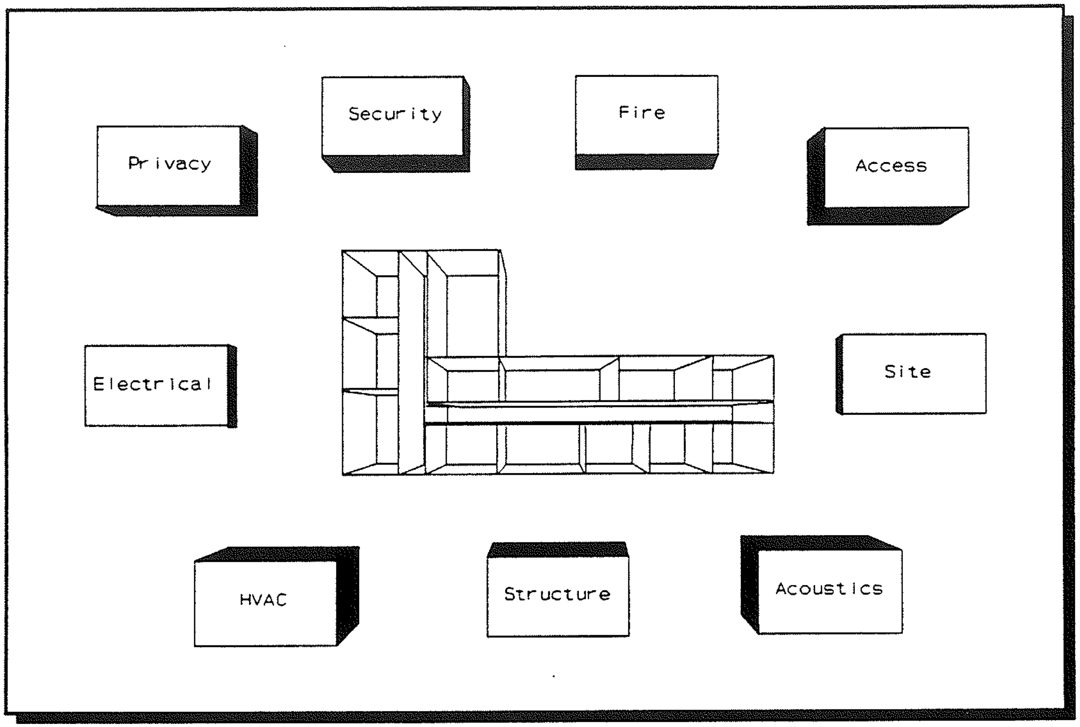


Figure 2-1: A sample of the multiple attributes impacting layout configurations

<u>Multiple Attributes</u>. The focus of initial layout generation systems on adjacency requirements highlights the deficiency of viewing a design in the context of a single interior attribute. Optimization in terms of a single attribute addresses the requirements related to that attribute, however, the resulting layout will be a trivial solution in terms of the remaining design program requirements. Layout evolution requires a designer to continually address **multiple design attributes** including privacy, occupant access, physical dimensions, and security (Figure 2-1, adapted from Pohl et al. 1988).

The impact of this issue on system development centers on the necessity to incorporate sufficient flexibility to concurrently address several design attributes. Rather than completing a configuration based on a single attribute, and subsequently analyzing the layout for compliance with the remaining attributes, the complete

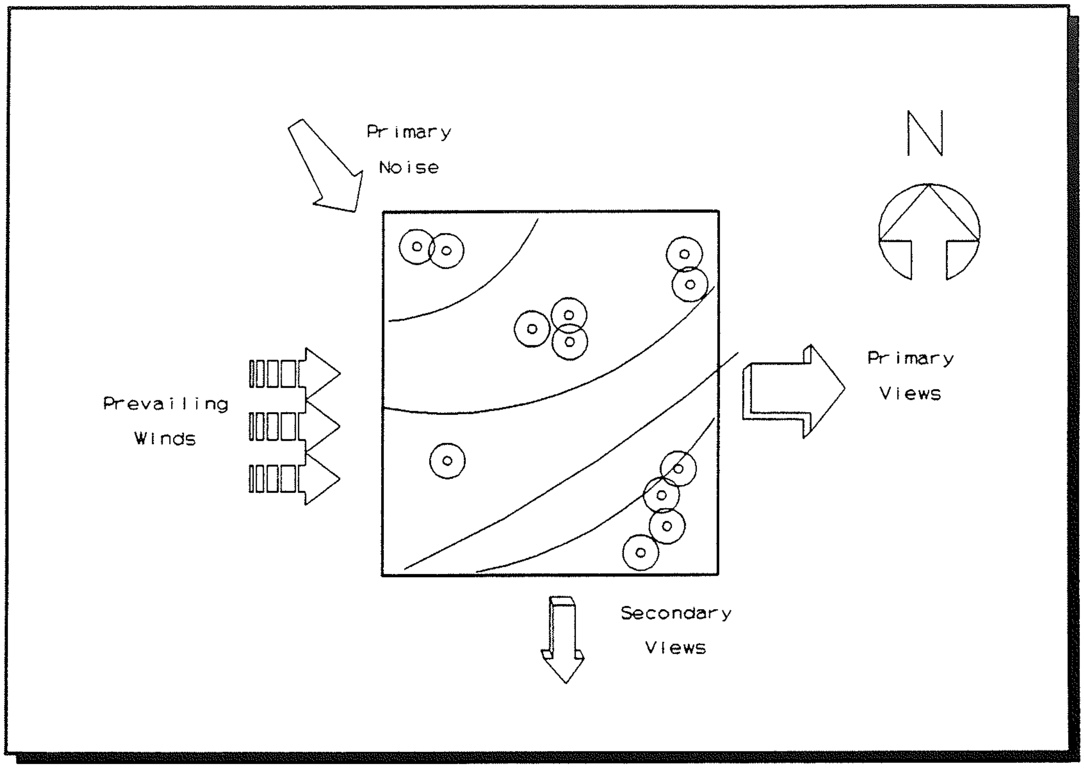


Figure 2-2: External layout influences require a system to contain orientation knowledge

spectrum of attributes should be addressed as placement options are generated. This will ensure that all attributes are integrated as a fundamental part of the final solution. Accomplishing this concurrent focus requires the inclusion of knowledge which will permit the system to reason about multiple attributes throughout the design process. Thus, a structure is required which facilitates utilizing this knowledge as it is warranted by the evolving solution.

<u>External Influences</u>. In addition to the multiple attributes impacting interior spatial relationships, configurations are impacted by external influences originating from the environment surrounding the layout (Krauss et al. 1970). These influences include site characteristics such as the buildable area and existing views, neighborhood context, and zoning considerations. Building types with an internal

efficiency emphasis such as warehouses or hospitals, may be designed with a minimum of attention to external influences, however, the majority of building types are significantly influenced by external environmental issues. These concerns influence the placement of spaces in relation to views on the site, pedestrian walkways, and traffic corridors. In the case of particular spaces such as executive offices, these influences play a primary role in dictating spatial locations.

The addition of external influences within layout generation systems presents similar requirements as those related to multiple attributes. Systems need the flexibility to reason about various external requirements, as well as, the flexibility to incorporate this reasoning capability in the problem solving process at the required time. Additionally, systems must recognize external concepts such as compass directions, to properly address site conditions. For example, given the simple site conditions illustrated in figure 2-2, systems need to distinguish between north and east to properly address requirements for primary and secondary views. The lack of this capability will restrict consideration of external relationships and requirements.

Design Knowledge. The layout generation process cannot be characterized entirely in terms of the previously discussed layout requirements. An additional element which plays an essential role in the process is the architectural knowledge a designer brings to bear on a layout problem. This knowledge represents the experience and formal training acquired over the course of a designer's career. The application of this knowledge provides the capability to develop layouts which transcend geometric manipulations to the application of established principles and preferences.

The impact of design knowledge on layout generation systems focuses on several aspects including representation, reasoning, and underlying control. In contrast to design requirements which are static elements on which design decisions are based, design knowledge is an active element incorporated to assist in decision making processes. This knowledge may determine the appropriate time to transfer system control to the designer, to transfer system emphasis to different decision making phases, or to transfer system emphasis to different layout stages. The determination of these decisions is a dynamic process dependent on the conditions and circumstances emerging in the layout generation process. Therefore, to facilitate this reasoning, systems require the capability to apply knowledge at various points within the layout generation process and in the various capacities in which it is needed.

The central impediment to achieving this reasoning capability is knowledge representation (Lenat and Guha 1990). Design attributes are amenable to both traditional representation paradigms such as files and databases, and knowledge-based paradigms such as frame hierarchies. In contrast, design knowledge specifically requires the use of knowledge-based paradigms such as rules and semantic networks, to capture the conditions under which the knowledge may be appropriately utilized. This representation requirement precluded this issue from being addressed in initial mathematical-based systems. However, advancements in artificial intelligence research have provided the fundamental representation paradigms necessary to include design knowledge in current and future layout generation systems.

Design Concepts. Design concepts are the underlying spatial ordering paradigms incorporated within layout configurations (Ching 1979). Concepts may be form driven such as linear, clustered, or radial (figure 2-3), or may evolve based on layout requirements. These concepts provide a configuration with a basis, or parti, from which to draw a central focus. The configuration thus assumes a coherent form and underlying rationale. In contrast, the lack of a design concept may result in a configuration appearing as a collection of haphazardly placed spaces lacking a unifying theme.

Design concepts vary from external influences and design attributes by requiring layout generation systems to incorporate knowledge about design themes, rather than design requirements. This knowledge provides the capability to generate

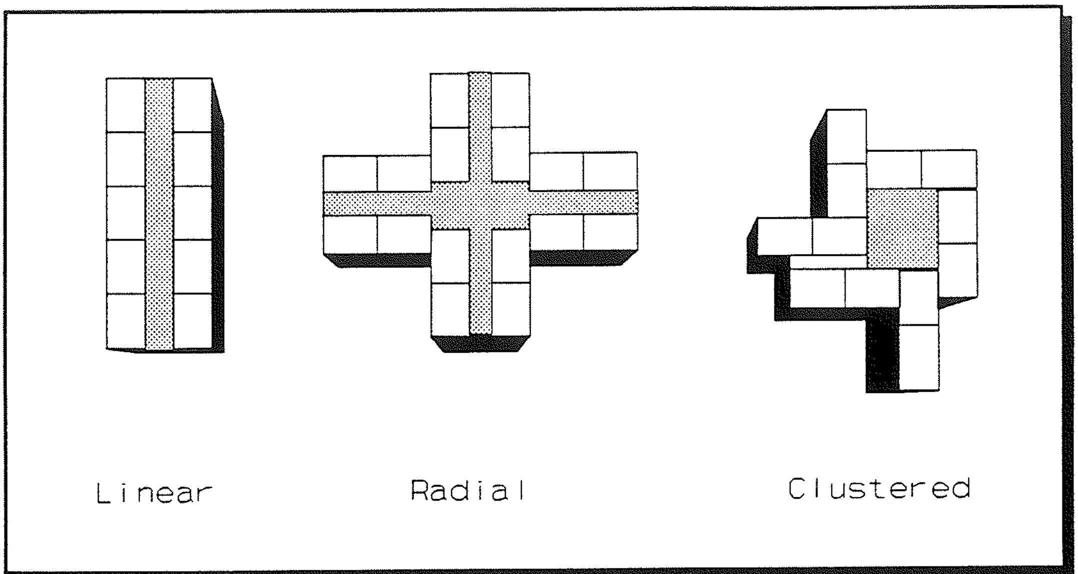


Figure 2-3: Examples of design concepts which provide a layout with an underlying organization

individual placement decisions which collectively reinforce the design concept. For example, if two spatial placement options exist within a layout incorporating a linear concept, but only one option adjoins the main axis, then the option adjoining the main axis should be chosen due to its reinforcement of the linear concept. This decision making emphasis varies from that related to the other design influences due to its overall design focus. Whereas design requirements provide a solid basis for determining placement option compliance, design concepts require decision making processes capable of determining the acceptability of placement options based on less stringent and less explicit constraints. Additionally, these decisions will have to be made within various design contexts to ensure that the concept remains the central layout emphasis.

Design Process. The architectural design process has been extensively studied by researchers in an attempt to document the fundamental processes used to create architectural entities from design programs (Hawkes 1976; Jones 1979; Hyde 1989;

Maher 1990). The results of these studies emphasize the existence of two primary processes within the architectural design context: an iterative design process focused on decision making, and an evolutionary design process focused on generating designs through multiple phases.

The former of these two processes, the design decision process, is the emphasis of this design issue. Research has revealed that designers use an analysis-synthesis-evaluation cycle to iteratively arrive at design decisions (Simon 1975; Akin 1978). Within this cycle, designers analyze the current design circumstance to generate further layout constraints, synthesize the information into partial design solutions, and evaluate the solutions for constraint compliance. A designer may repeat each of these phases several times in an effort to analyze, comprehend, and finally, make a design decision addressing the various problem issues (Pohl et al. 1988). These repetitions provide an opportunity to generate a solution, or group of solutions, through a process of iterative refinement (i.e., the designer iteratively refines the configuration until a desired solution is achieved).

The importance of the iterative design process to layout generation system development is evident in the documented use of the process by designers (Akin 1988; Eastman 1970). However, the importance of the process is further emphasized through an examination of the design processes incorporated in initial layout generation systems. Within these systems, a minimum degree of analysis, if any, was conducted prior to synthesizing design solutions (Grant 1983a; Grant 1983b). Evaluation was based on narrowly defined parameters such as spatial distances (Mitchell 1977). The emphasis was on synthesizing solutions based on designer provided requirements. This emphasis negated the potential for examining alternative decisions based on different analysis scenarios, and limited the value of evaluation feedback. Based on this restricted capability, the importance of the three phases becomes apparent in terms of providing valuable information and design opportunities. Therefore, a layout generation system needs the flexibility and expanded capabilities to perform extended analysis functions, in-depth evaluations, in addition to synthesis procedures.

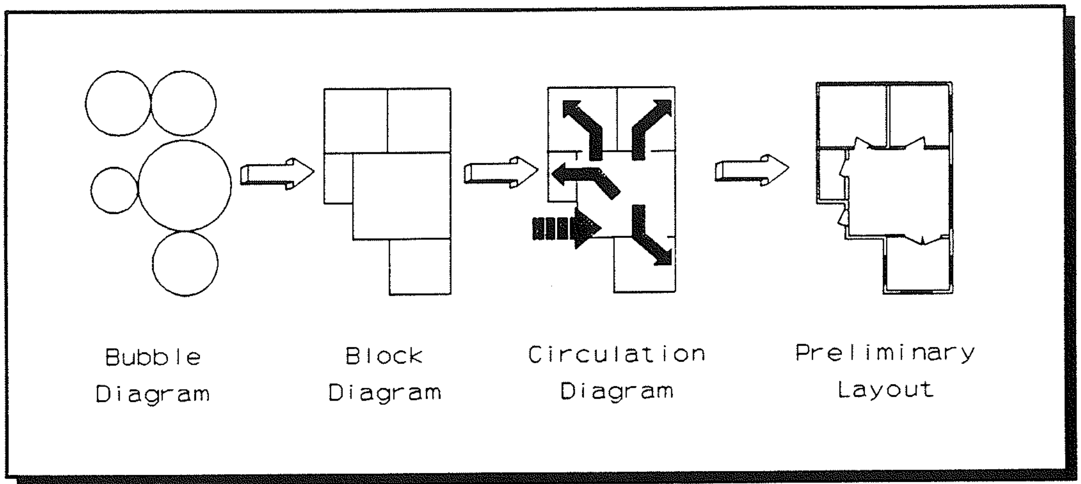


Figure 2-4: A model of the multiple stages through which a layout evolves within the conceptual and preliminary design phases

<u>Multiple Design Stages</u>. The second layout generation process stresses the evolutionary process of creating entities such as layouts or buildings, through multiple design phases. In each phase, designers develop layouts by refining potential configurations through several stages, each of which addresses a greater number of design details than the last (Harrigan and Harrigan 1979). This process permits designers to address design issues at various levels of abstraction (Galle 1986). At the highest level of abstraction, the designer addresses issues at a conceptual level by considering spaces as amorphous shapes. At a lower level of abstraction, the designer addresses detailed design issues including the placement of entry ways and circulation corridors. Through multiple iterations between these abstraction levels, designers vary the level of detail at which individual issues are addressed until a solution is obtained at a uniformly detailed level.

Figure 2-4 portrays a model of the multiple stages a designer moves through during the layout generation process. At the conceptual diagram stage, the designer emphasizes spatial relationships and design concepts while leaving the spaces as amorphous shapes. These shapes are then refined to blocks and finally to actual dimensions through the next several stages. Concurrently, the circulation patterns, access ways, fenestrations, and other design attributes are brought in at increasing levels of detail and specificity. This process is not a linear process. Several times during layout evolution, designers will return to earlier stages of the process to either gain a better understanding of an issue, or create layout alternatives. Similarly, designers may proceed to a further level of detail in a specific domain to conduct further evaluations of the solution feasibility.

The impact of this process on the development of layout generation systems centers on the scope and objectives of individual systems. Systems emphasizing the generation of layouts at the middle stages of abstraction (i.e., generating layouts between block and circulation study levels), eliminate opportunities to explore design issues at conceptual levels. Designers must have definitive information regarding spatial areas and dimensions prior to using the system. Currently, in the author's opinion, it is not practical to develop systems incorporating the entire range of design stages, due to the vast diversity of knowledge required at each abstraction level[3]. The lengthy effort required to capture this volume of knowledge falls outside the objectives of the majority of research efforts. However, a system may emphasize the development of layouts at a given level, or a narrow range of levels, by capturing the appropriate knowledge. The focus on these stages determines the system potential for supporting layout abstraction, and establishes the scope in terms of supporting the overall evolutionary process.

---

[3] This opinion is based on the author's studies and personal communications with other researchers. It is not intended to be a research hypothesis or a proven axiom.

Computational Issues

Whereas design issues focus upon the aspects of layout generation associated with architectural design, computational issues emphasize the issues associated with generating layout configurations from a system viewpoint. The computational issues include user control, multiple configurations, and expandable theory. The following descriptions introduce these issues together with a brief discussion of their impact on layout generation system development.

User Control. The evolution of layout generation systems has witnessed a vast array of approaches to user control. These approaches range from excluding the user from all aspects of layout generation except initial data input (Henrion 1978), to completely interactive systems requiring the user to provide data and input at every layout generation decision point (Akin, Dave, and Pithavadian 1988). The approach applied within an individual system significantly impacts its role in the layout generation process. At one extreme, if the designer is excluded from the process, then the system acts as an independent design generator. In contrast, if the designer is in complete control, then the system serves as an electronic sketch pad on which to evaluate alternatives and clarify issues. In between these two extremes, the system serves as a design assistant, giving various degrees of designer feedback.

Coinciding with the issue of user control is the subissue of layout manipulation. This manipulation may range from the system producing a layout and allowing no designer manipulation, to allowing a designer to manually alter the configuration into an entirely different entity with the assistance of system feedback. The implementation chosen for an individual system dictates the flexibility to generate alternatives based on initial solutions.

The implementation approach selected for each of these control issues will impact the character, flexibility, and underlying purpose of a layout generation system. In terms of user control, if the intent is to create a user controlled system, then interaction capabilities including graphical depictions, user input, system messages, and additional devices are required to provide the information necessary to make appropriate layout decisions. In terms of layout manipulation, if a system allows the user to alter generated solutions, then the modes of manipulation need to be considered. Alteration modes may focus on any combination of direct layout manipulation, attribute manipulation, or requirement manipulation. The selection of these modes directly impacts the system interface requirements.

Multiple Configurations. The numerous influences within a design program provide the potential for multiple configurations to exist which each satisfy the design program requirements. This possibility exists due to the influences being primarily soft constraints. Whereas hard constraints such as minimum space dimensions, must be complied with in the configuration, soft constraints such as the desirability for a space to be located in a public zone, may be relaxed to accommodate other layout considerations. Soft constraints represent designer or owner preferences which may not always be accommodated due to conflicting requirements. In these circumstances, trade-offs must be made between the soft constraint requirements. Thus, through various combinations of constraint trade-offs, spaces may be located in several possible locations. The aggregation of these location possibilities results in numerous potential configurations for each design program specification.

The computational issue to be addressed in response to this multiple configuration potential, is how to reduce the number of partial configuration possibilities existing at each generation step. The number of possibilities continues to expand as the layout evolves due to the accumulation of partial solutions. As each space is added to the configuration, the influence trade-offs present several location opportunities, each of which represents a new partial solution to the problem. An early study projected that nine spaces containing only adjacency and area requirements, could generate approximately 250,000 configuration possibilities if every partial layout was retained and expanded throughout the layout process (Mitchell, Steadman, and Liggett 1976, 46).

In response to this issue, researchers are exploring options to reduce the number of partial configurations including shape grammars (Chase 1989), intermediate evaluations (Flemming et al. 1988), and expert systems (Coyne et al. 1990). The common theme underlying these approaches is to either eliminate less viable options after they are generated, or prevent the generation of less viable placement options as the space is being placed.

<u>Expandable Theory</u>. The architectural domain comprises a wide range of building types from residential dwellings containing less than ten spaces to medical facilities containing hundreds of spaces. The disparity between these building types results in a fundamental issue related to system applicability. A system restricted to solving limited domains such as efficiency apartments, can incorporate a layout generation theory developed to solve that particular problem. However, the application of the same theory to the medical facility problem may prove to be inadequate due to the increased number of spaces, relationships, and issues.

This disparity in building types, and their associated complexity factors, influences the expansion of layout generation theories. Specifically, a determination is required during early development stages in respect to the eventual system application. If the intent is to create a general purpose layout generation tool, then this will require a vastly different research approach than an intent to develop a system focusing on a narrow range of layout generation problems. The development of a theory applying to problems with a limited number of spaces and relationships may need significant alteration if the intent is subsequently changed to address domains with greater complexity factors. This acknowledgement is a necessary component of system development and a rudimentary stage in defining the system scope.

CAADIE Definition

The definition of the design and computational issues influencing system development provides a basis for reducing the ambiguity present in previous definitions of the layout generation problem. The following definition reduces the possibility to modify the problem scope, by explicitly specifying the issues comprising the layout generation problem.

> Layout generation is the process of placing a set of spaces in a specified area, where the resulting configuration incorporates the multiple attributes, external influences, and design concepts enumerated in the design program.

The layout generation issues provide a basis to analyze systems according to their use of iterative design processes, design knowledge, and multiple design stages to address layout requirements. Additionally, the computational methodologies used to achieve this implementation can be analyzed in terms of user control, multiple configurations, and expandable theories. The following chapter introduces various approaches to these issues, together with analyses of the approaches based on the defined layout generation issues.

CHAPTER 3

REVIEW OF CURRENT WORK

Research efforts addressing architectural layout generation encompass a diversity of computational approaches. These efforts have evolved from the use of mathematical approaches for spatial distance optimization, to the use of artificial intelligence paradigms emphasizing architectural reasoning. Comprehensive examinations of specific segments within this research domain have previously been provided by Grant (Grant 1983a), Henrion (Henrion 1978), and Schmitt (Schmitt 1988a). This chapter diverges from these approaches by focusing on a broader classification of layout generation research. This classification emphasizes three research areas: algorithmic-based systems, knowledge-based systems, and integrated systems.

The first part of the review provides descriptions and analyses of research efforts within the three primary categories. These descriptions highlight the advantages and disadvantages associated with each approach in terms of the previously defined layout generation issues. The second part of the chapter provides a further analysis of knowledge-based design research efforts directly influencing the CAADIE research effort.

Algorithmic-Based Systems

The initial drive to develop architectural layout generation systems focused on the augmentation of existing mathematical algorithms to examine the spatial synthesis capabilities of computers. The underlying objective focusing on the generation of spatial organizations based on requirements provided by the user. Results from these efforts have provided the foundation from which research into layout generation algorithms continues to the present. This continued research has expanded the use of algorithms beyond the early synthesis studies to algorithms which address the complete analysis-synthesis-evaluation design process.

The algorithmic approaches explored during the last three decades of layout generation research have succeeded in creating a diversity of variations and specializations. Within this diversity, several approaches including quadratic assignment, graph theory, and rectangular dissections, have evolved into the fundamental core of algorithmic research efforts.

Quadratic Assignment Systems

The earliest research in automated layout generation built upon optimization work conducted in operations research and economics. This layout generation research emphasized the use of evaluation functions to optimize cost considerations within a given facility (Koopmans and Beckmanns 1957). Specifically, the cost consideration was evaluated based on spatial distances. This factor was chosen based on the reasoning that spaces requiring materials and people to be moved between them should be placed as close together as possible. The quadratic assignment formulation used to determine this optimization is defined as follows:

> Total Circulation Cost =
> The sum of the circulation costs obtained for each space, where the circulation cost for each space is obtained by taking the sum of the circulation costs between the space and every other space. The circulation cost between the spaces is given by multiplying the cost per distance unit between two spaces with the distance between the two spaces (adapted from Mitchell 1977, 426).

Figure 3-1 illustrates a typical quadratic assignment example. In this figure, the circulation cost for space A is determined by (1) multiplying the cost per distance unit factors between A and the other two spaces with the respective distances between A and the other two spaces; and (2) taking the sum of these products. This

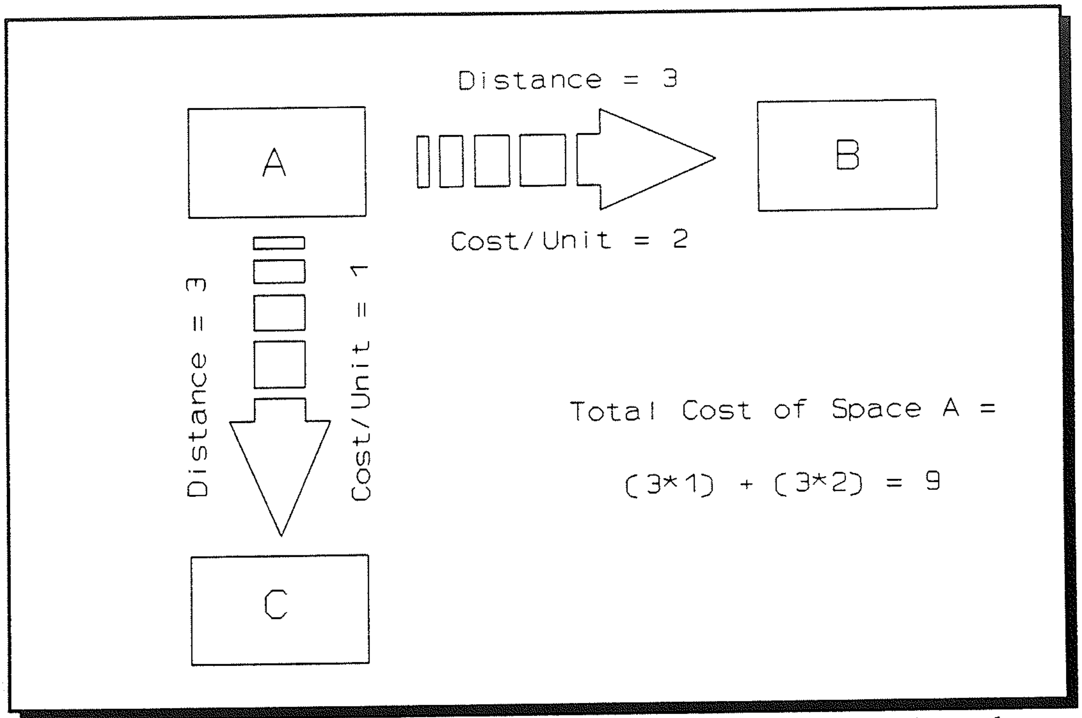


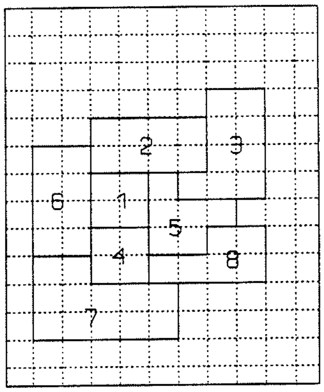
Figure 3-1: An example of the quadratic assignment approach to layout optimization

procedure subsequently is used to obtain the individual circulation costs for each remaining space. The total circulation cost is then calculated as the sum of the individual totals.

This cost evaluation technique formed the basis for two ground breaking layout generation systems. The first of these systems, CRAFT, was a permutational based system (Armour and Buffa 1963). CRAFT required users to provide an initial layout and cost per distance unit factors for each space. The system evaluated the initial layout and provided a base cost evaluation. The various layout permutations were then evaluated by transferring pairs of spaces within the layout. This procedure continued until the system generated all of the potential permutations from the initial layout. The layout with the optimal overall cost factor would then be selected as the "best" alternative.

Area Requirements

Space	Area
1	100
2	200
3	225
4	100
5	150
6	200
7	300
8	150



Adjacencies

Space	Req's
1	2,4,5
2	1,3,6
3	2,5,8
4	1,6,8
5	1,3
6	2,4,7
7	6
8	3,4,6

Figure 3-2: The grid allocation method introduced in the CORELAP research effort

The second system, CORELAP, represented an initial group of research efforts into the development of additive systems (Lee and Moore 1967). These systems developed layouts by sequentially combining individual spaces. CORELAP did not require an initial layout, however, the cost per distance unit factors were still required. A series of potential layouts were generated based on given adjacency requirements, the required areas for each space, and a grid on which to place the spaces (figure 3-2). The layout options were evaluated based on the quadratic assignment formulation, and the resulting optimum layout was presented to the user.

The underlying permutational and additive concepts evident in CRAFT and CORELAP respectively, formed the basis for layout generation research throughout the next decade. Researchers including Willoughby (Willoughby 1975) and Cinar (Cinar 1975), extended the CRAFT work to address issues such as 3-D design and building form. Researchers including Mitchell (Mitchell, Steadman, and Liggett 1976) , Liggett (Liggett and Mitchell 1981), and Weinzapfel (Weinzapfel and Handel 1975), extended the CORELAP work to address various issues within the layout generation problem.

Advantages

The quadratic assignment methodology addressed two significant layout generation issues: design process and expandable theory. In terms of design process, the methodology introduced an evaluation component into the automated layout generation process. Rather than focusing exclusively on the synthesis of configurations, the quadratic assignment theorem provided the opportunity to evaluate solutions based on a defined set of criteria. This evaluation represented the first implementation of a process to determine layout requirement compliance. Additionally, the evaluation component exemplified the first attempt to present designers with a ranked order of configuration solutions based on designer provided requirements.

The second quadratic assignment contribution addressed the expandable theory issue. The mathematical basis provides a size independent mechanism for rating individual placement options and overall layouts. Since the quadratic assignment procedure uses a single set of equations to evaluate distances between spaces, the number of spaces is not a consideration in terms of approach limitations. Given any set of spaces, a configuration containing a minimum distance rating is achievable through the quadratic assignment equations. Subsequently, the system can portray the configuration as an aggregation of individual mathematical solutions which collectively form the final organization rationale.

Disadvantages

In contrast to these advantages, the quadratic assignment approach contains numerous limitations which ultimately led to its abandonment as a general purpose layout generation methodology. Notable among these limitations, is the failure to address the multiple attribute issue. The underlying assumption that circulation is always the overriding design attribute, unrealistically simplifies the layout generation problem. As discussed previously, many attributes influence any given layout. Therefore, outside of circulation constrained problems such as industrial plant design, the quadratic assignment approach generates layout solutions which ignore the complex attribute interrelationships.

Graph Theory

The spatial relationship emphasis in algorithmic-based research, prompted design researchers to explore alternative methods for representing these relationships. The format selected by several researchers centered on graph theory formalisms. In early papers by Grason (Grason 1970b) and Steadman (Steadman 1976), researchers documented the potential for portraying relationships as links within a graph representation. These researchers suggested that the application of this formalism would permit established graph theory algorithms to be implemented within layout generation systems.

Figure 3-3 illustrates this representation scheme. In this illustration, a set of adjacency requirements is initially provided by the user (figure 3-3a). Based on these requirements, a graph is constructed where the spaces are represented as nodes and the adjacency requirements are represented as links between the nodes (figure 3-3b). The graph illustrated, referred to as a requirements graph (Steadman 1976, 97), indicates the existence of potential configurations based on the condition of planarity (i.e., the graph contains no intersecting links). Given this condition, graph theories

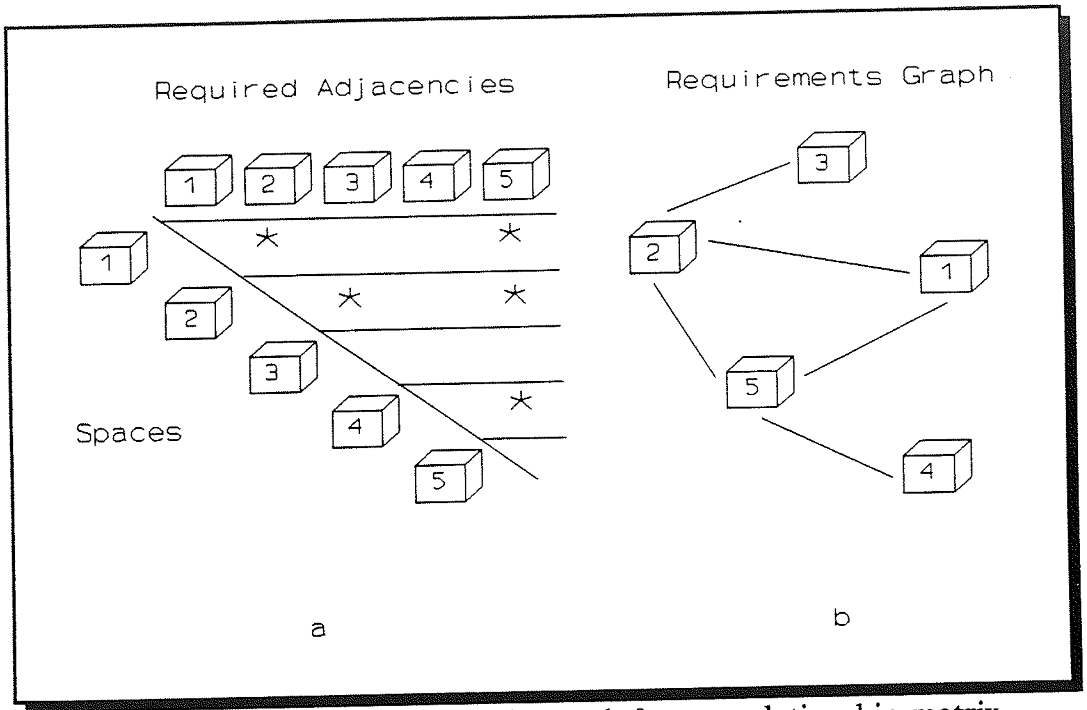


Figure 3-3: Generating a relationship graph from a relationship matrix

prove that all relationships can be accommodated in a 2-dimensional plane. This condition permits systems to analyze requirements and determine the existence of solutions prior to the synthesis phase. Given the planar requirements graph, a series of potential layouts may be generated which satisfy the spatial requirements (figure 3-4a). Finally, a second graph representation, referred to as a dual graph (Steadman 1976, 98), characterizes the layout adjacencies and common walls by distinguishing the north-south adjacency links from the east-west adjacency links (figure 3-4b).

Research efforts founded on this theory have resulted in several layout generation system implementations. Notable among these, are implementations by Grason (Grason 1970a), Baybars and Eastman (Baybars and Eastman 1980), Hashimshony (Hashimshony and Roth 1986), and Rinsma (Rinsma 1988). Although these efforts comprise different methodologies to generate layout alternatives, each incorporates a general methodology based on generating graphs from relationship requirements and subsequently, enumerating layout alternatives.

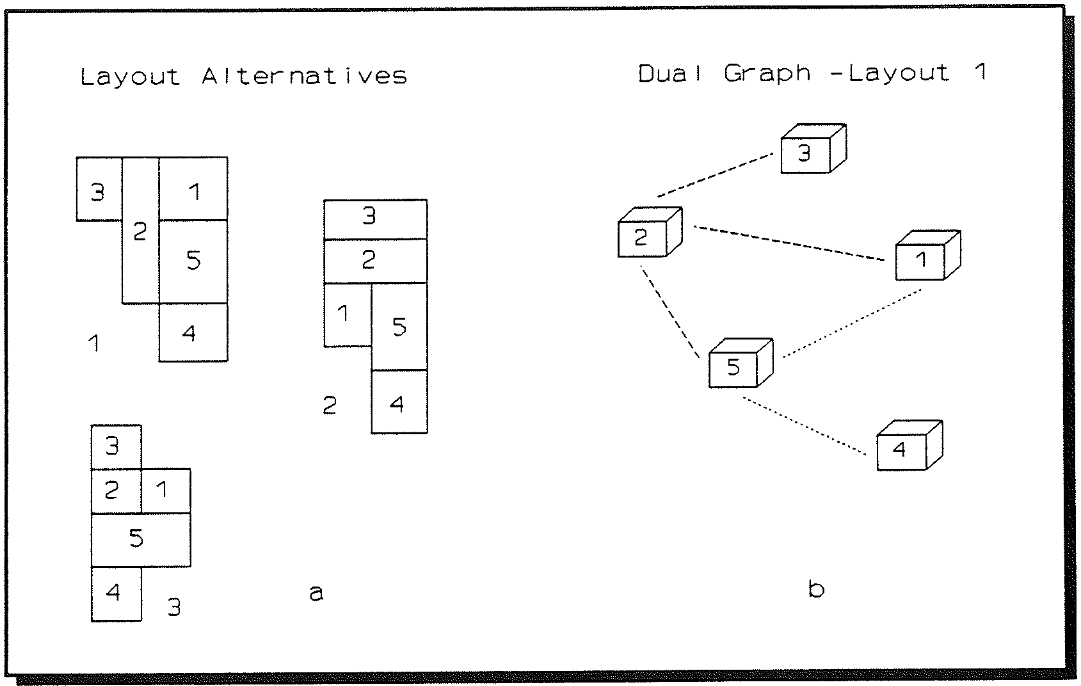


Figure 3-4: Layout alternatives and a dual graph representation generated from the relationship graph in figure 3-3

Advantages

Similar to quadratic assignment implementations, graph theory implementations address the expandable theory issue through their mathematical basis. Additionally, the graph theory approach addresses two further layout generation issues: multiple attributes and multiple stages. The ability to include multiple attribute requirements represents a notable advancement, based on the possibility to generate configurations from factors other than circulation. Any design attribute capable of being represented by a relationship graph, can be used as a configuration basis. For example, external access requirements such as daylighting and views, can be represented as adjacencies to the appropriate external directions.

The resulting configuration will incorporate the daylighting requirements as the layout basis. In this way, generated configurations overcome the circulation constrained boundary imposed by optimization techniques.

The second issue, multiple stages, is facilitated by the consistent representation associated with dual graphs. Graph theory formalisms restrict requirement representations to the portrayal of spatial relationships. However, although the physical attributes of spaces change through the layout generation stages, spatial relationships such as adjacencies, remain as non-geometric entities. Therefore, the same representation may be used to characterize relationships in both conceptual design and detailed design stages. Furthermore, this commonality provides an initial basis for developing layout generation systems capable of evolving layouts through multiple design process stages.

Disadvantages

The previous advantages continue to create interest in a graph theory basis for layout generation. However, graph theory representations contain an inherent limitation in their recognition of non-adjacency based issues such as design concepts and economic viability. These issues require information and knowledge beyond spatial relationships. For example, design concepts require information related to the application of design principles, and economic analyses require external environment input. This form of information cannot be captured in the fundamental graph representation. Thus, the expansion of the approach to a greater realm of design issues requires secondary representation paradigms. This limitation has resulted in current approaches utilizing graph representations as one component in a hybrid representation format which retains a mathematical basis while expanding the spectrum of addressed issues.

Rectangular Dissections

The final area of research in this category originated with examinations of geometric algorithms, referred to as rectangular dissection algorithms, developed to divide rectangular regions into smaller rectangles. Layout researchers drew comparisons between this process and the process of dividing up a given floor area into individual spaces. Initial efforts emphasized the use of this dissection process to enumerate configurations which could be generated from a given floor area, a set of required spaces, and a set of required adjacencies between these spaces (Mitchell, Steadman, and Liggett 1976).

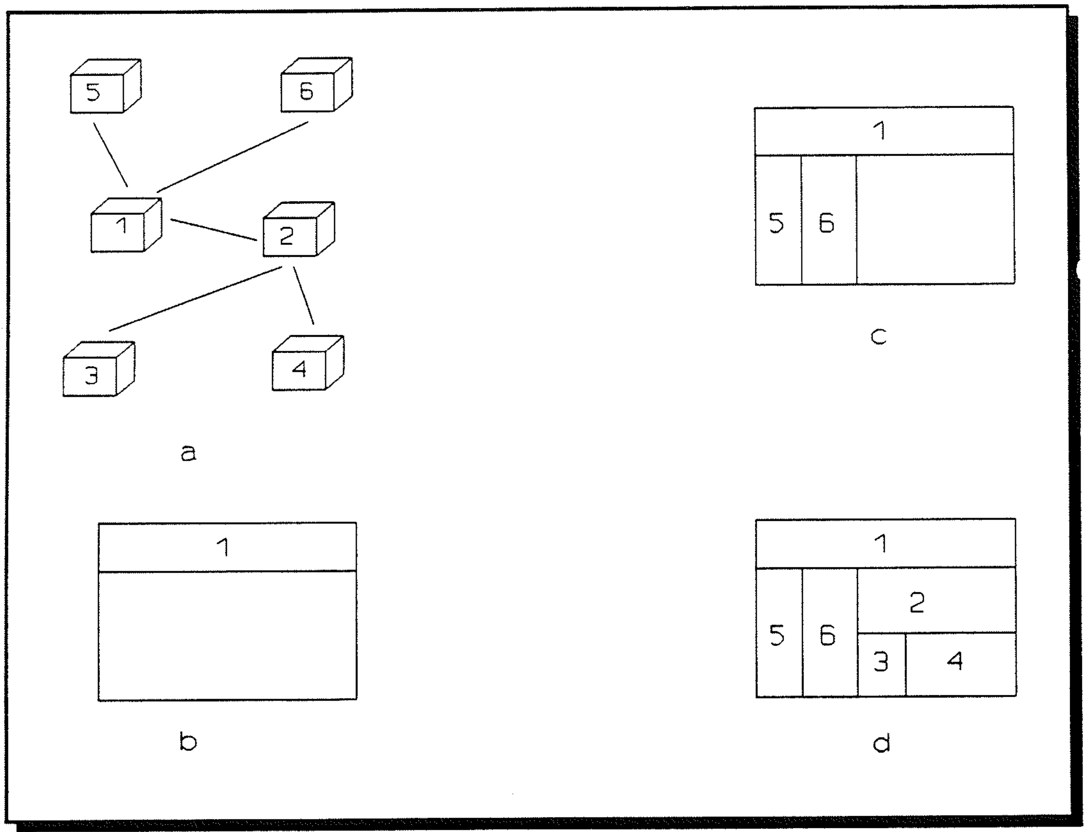


Figure 3-5: The generation of a layout alternative through rectangular dissection

Figure 3-5 illustrates the rectangular dissection concept applied to the generation of a single configuration. Initially, a relationship graph is generated based on an adjacency relationship matrix (figure 3-5a). The algorithm then randomly selects one node from the graph as the first space to generate. After selecting this node, the rectangle is dissected into two sections, the first contains the area required for the first node, the second contains the remaining area required for the configuration (figure 3-5b). The remaining area is then dissected into two sections, one containing the total area for spaces 2, 3, and 4, the other containing the area for spaces 5 and 6 (figure 3-5c). This process continues until all required spaces have been allocated (figure 3-5d). In this same manner, alternative configurations are generated by selecting alternative nodes as starting points for the dissection process.

This geometric enumeration process formed the foundation of several research efforts including one by Liggett focusing on the enumeration of all possible configurations for a given dwelling type such as a small apartment (Liggett 1972). This effort produced promising results, however, one limitation encountered was the potential for large numbers of configurations to be generated from a single graph. For example, given ten spaces, ten different starting points are available. From each of these starting points, the option exists as to which branch of the graph to initially pursue. In the same manner, each subtree in the graph presents the same options. Thus, numerous configurations could be generated from a single graph. Subsequent research efforts are exploring this limitation through extensions to the dissection formalism (Rinsma 1988), and the addition of complementary techniques such as rule-based evaluation (Flemming 1988).

Advantages

The rectangular dissection methodology retains the expandable theory advantage evident in the previous mathematical based approaches. In addition, this approach addresses the multiple configuration and multiple stage issues. In terms of the former issue, the rectangular dissection formalism introduced the first alternative to the CORELAP grid allocation method. The ability to generate multiple configurations based on different relationship graph nodes permits the methodology to create a number of layout alternatives from a single set of relationships. The selection of these starting nodes results in solutions which vary significantly in their geometric configurations, while retaining compliance with their relationship requirements. Thus, the system presents a number of fundamentally different alternatives, each of which is viable according to the spatial relationship requirements.

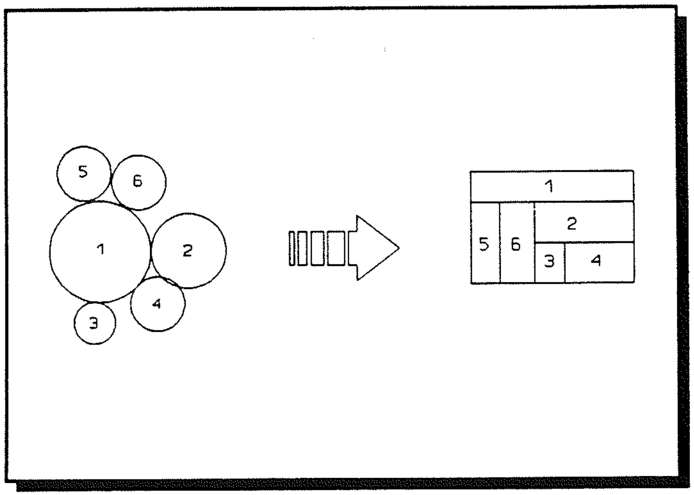


Figure 3-6: The transition from bubble diagrams to rectangular dissections provides a consistent, underlying representation

Complementing this advantage, is the ability to utilize rectangular dissections to evolve solutions from bubble to block diagrams. Given the association between relationship graphs and rectangular dissections, space relationship graphs can represent initial bubble diagrams (figure 3-6). Once these diagrams are completed, the rectangular dissection formalism may be invoked to generate initial block diagrams. Through this methodology, the formalism provides a basis for evolving configurations between several conceptual design stages.

Disadvantages

In addition to retaining the advantages evident in previous mathematical approaches, rectangular dissections retain the disadvantages including the lack of capability to address non-adjacency based relationships. However, this approach highlights a further deficiency, the lack of user control. The automated transformation of relationship graphs into rectangular dissections eliminates designer participation. The designer is acknowledged solely as an initial input source. This approach limits the potential for designer participation, and disregards the experience and intuition a designer brings forth to each design problem. Ignoring this experiential knowledge reduces the system capability to incorporate all available information in potential system solutions, and reduces the system capacity to provide design assistance based on user initiated requests.

Knowledge-Based Systems

The significant development of knowledge-based system technology over the last ten years has resulted in an equally significant impact on layout generation research. The introduction of this technology presented researchers with a fundamentally different paradigm for addressing the layout generation problem. Whereas previous research focused on the development of algorithms to simulate the layout generation process, knowledge-based paradigms provided the capability to focus on design process reasoning. Specifically, knowledge-based systems presented an approach to capture the rules of thumb, or heuristics, designers utilize throughout the layout conceptualization process. The significance of this advancement is highlighted by the following criticism of algorithmic-based systems:

> One must realize, however, that the operations performed by designers are ill-understood, at least at the level of precision and explicitness needed if they are to be expressed through a computer program.

> Except for well-understood, special cases, no theories exist that would lead easily to programs able to perform interesting design tasks (Flemming 1988, 94).

The further development of knowledge-based system technology provided expanded reasoning and knowledge representation capabilities for design researchers. These capabilities began to address the limitations encountered in algorithmic-based systems in terms of shape representations, the use of non-topological attributes, and the use of flexible reasoning paradigms for design process control. Based on these advances, researchers have explored various applications of this technology in layout generation prototypes. The following sections highlight three categories of pivotal knowledge-based system research: expert systems, shape grammars, and layout analysis systems.

Expert Systems

The initial effort to incorporate knowledge-based paradigms in layout generation systems emphasized the use of expert system formalisms. In these systems, design knowledge is represented within the condition-action formalism of rules. The rules capture the specific conditions under which designers reach decisions for a limited design domain, together with the actions a designer takes when these conditions are present. For example, the following rule captures a design heuristic focusing on the placement of two spaces with a required adjacency:

IF	a space has been placed in a configuration,
AND	the next space to be placed contains an adjacency requirement with the first space,
AND	an available placement position exists
THEN	place the next space in the available position.

Based on initial studies in limited design domains such as office design and kitchen design, researchers demonstrated the potential of using a rule-based paradigm to capture sufficient designer knowledge for layout generation applications.

Subsequent expert system applications have focused on the development of rules to formalize the use of standard configurations and to study architectural reasoning processes.

Design Methodology

Layout conceptualization processes and methodologies have received notable attention since the introduction of "systematic design methods" thirty years ago. These studies analyzed design processes with the intent to introduce more specific and objective design methods (Henrion 1978). Based on results from researchers including Alexander (Alexander and Poyner 1970), Simon (Simon 1975), and Akin (Akin 1978), general rules and methodologies were developed to assist in objective layout development. Directly evolving from this research has been a series of efforts to develop expert systems incorporating these design methodologies.

A representative result of these efforts has been the development of a heuristic-based system referred to as HeGeL (Akin, Dave, and Pithavadian 1988). This system includes design rules to assist designers in addressing both layout problems and layout alternative generation. The system uses heuristics to provide designers with a limited number of design solution and process options for each circumstance encountered in the layout generation process. For example, when the placement of a specific space is constrained by attributes such as daylighting or adjacencies, the system presents the designer with a set of design process options such as focusing on space relationships or space attributes, which are appropriate for generating placement options.

Prototype Refinement

The use of rules to generate layouts through the application of standard configurations received considerable attention due to its basis in design methodology research. An argument put forth by both CAD researchers and design methodology researchers including Jones (Jones 1979), Wade (Wade 1989), Oxman (Oxman 1990), and Gero (Gero 1990), centers on the frequent use of previous design experiences and knowledge in design processes. These researchers contend that designers make significant use of previously developed solutions when addressing new problems. This use of previous design information includes the adaptation of design concepts, design methodologies, forms, and goals. As designers gain more experience in a given area, successful solutions to previous design problems become prototypes for future problems. Once these prototypes are developed, a designer rarely develops new prototypes due to the extensive knowledge which exists in previous prototypes. Thus, the act of designing does not focus exclusively on creating original design components, rather, design focuses on creating new configurations from existing components and applying them to current situations. Oxman summarizes this view in the following statement:

> It is an assumption of our work that design is in fact, a dynamic process of adaptation and transformation of the knowledge of prior experiences in order to accommodate them to the contingencies of the present (Oxman 1990, 18).

This design argument constitutes the basis for current research in the development of prototype refinement systems. These research efforts utilize rules to capture the design expertise required to determine the relevance and potential use of particular prototypes. To illustrate the general implementation of this concept, figure 3-7 demonstrates the selection of a kitchen location for a residential dwelling. In this example, the system contains prototype configurations for kitchen locations and selection rules such as the one illustrated, for selecting the appropriate prototypes. Based on given input specifications, the appropriate selection rule is used to determine a desirable kitchen location.

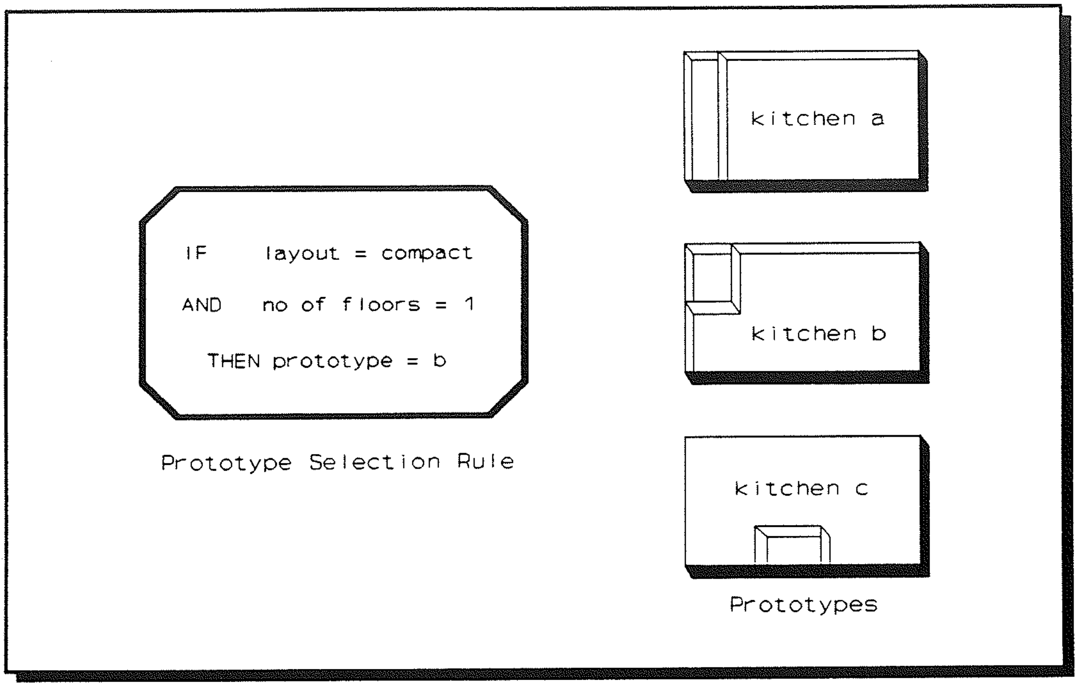


Figure 3-7: Prototype refinement rules select appropriate prototypes based on layout conditions

In a recent prototype refinement strategy implementation by Oxman and Gero, designers enter site conditions, access requirements, and additional specifications, which subsequently meet conditions in specific prototype selection rules (Oxman and Gero 1988). These rules determine the standard configurations for specific prototypes in the developing layout. The combined effect of firing individual prototype selection rules results in a suggested layout configuration. Extensions to the prototype refinement strategy by these researchers and others, are continuing to address issues such as improved prototype selection, increased attribute representation for each prototype, and the generation of new prototypes.

Advantages

The introduction of expert systems into the layout generation domain created a notable impact on the way researchers addressed several issues including multiple attributes, external influences, design concepts, and design knowledge. In the first three instances, this impact centered on the possibility to address attributes and issues beyond those related to adjacency relationships. For example, attributes such as acoustics and building form could be addressed through the capture of expertise related to these areas. The primary influence on the design knowledge issue diverged from the generation of layouts to focus on layout generation research methodology. Through the introduction of design knowledge, layout generation research has altered from the creation of geometric manipulation algorithms to the study of how designers utilize knowledge during the design process. Capturing the knowledge behind this process provides an opportunity to place areas of expertise within layout generation systems. The extended set of issues addressed by this knowledge expands the rationale behind layout generation decisions, and thus, results in solutions based on a greater diversity of design program requirements.

Knowledge Limitations

The introduction of design knowledge represents one of the greatest impacts on layout generation research, however, it also poses one of the most difficult questions for design researchers. The initial success in applying expert systems to limited design problems created an assumption that by identifying enough design knowledge, a system could emulate significant segments of a designer's process. Subsequently, efforts to identify this knowledge are becoming increasingly prevalent. The question emerging from these efforts is how much knowledge should be captured, and is possible to capture, within a design system. Answering this question relies on several issues:

- Given that designers continually build upon their experience, is it possible to capture a fundamental set of design knowledge.
- Given the cognitive limitations of designers to concurrently address multiple issues (Simon 1981), should this limitation be emulated within design systems.
- Is it possible, and desirable, to identify and capture the experience influencing designer preferences within a static rule base (i.e., a rule base that does not change in relation to the designer's expanding knowledge).
- Given the computational strengths of computers, should specific layout generation tasks remain algorithmic processes within the realm of system control.

These questions remain to be answered by current layout generation research efforts. As discussed in the opening chapter, the CAADIE project emphasizes an approach combining the strengths of the computer and the designer. Thus, it is not the intent of the project to capture all designer knowledge. Rather, the intent is to capture the reasoning and decision making capabilities permitting designers to address various design influences.

Shape Grammars

The use of forms and shapes as design vocabulary components has been associated with architecture, and designers in particular, for centuries. Examples of these vocabularies have been traced at least as far back as Greek Empire temple designs (De La Croix and Tonsey 1980, 120). Based on the history and prevalence of these vocabularies, design researchers have attempted to formalize methodologies from which an individual designer's use of forms and shapes could be defined and reproduced by other designers. Studies by March and Steadman into formal geometric manipulation theories demonstrated that a formalism could be developed to generate geometric configurations through a set of geometric transformation rules

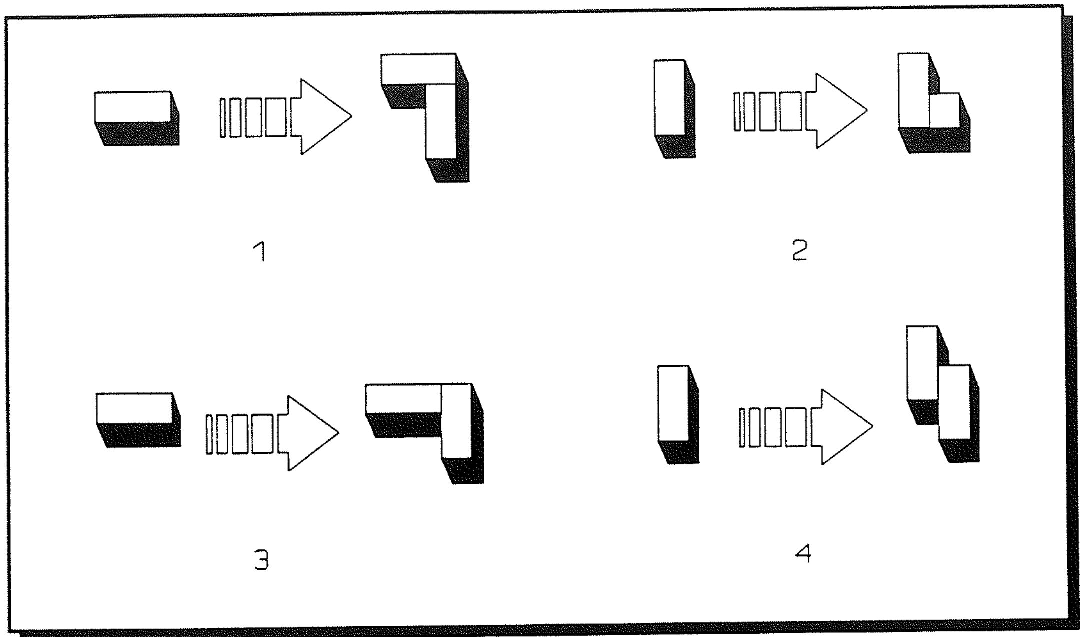


Figure 3-8: A shape grammar vocabulary defined by four shape grammar rules

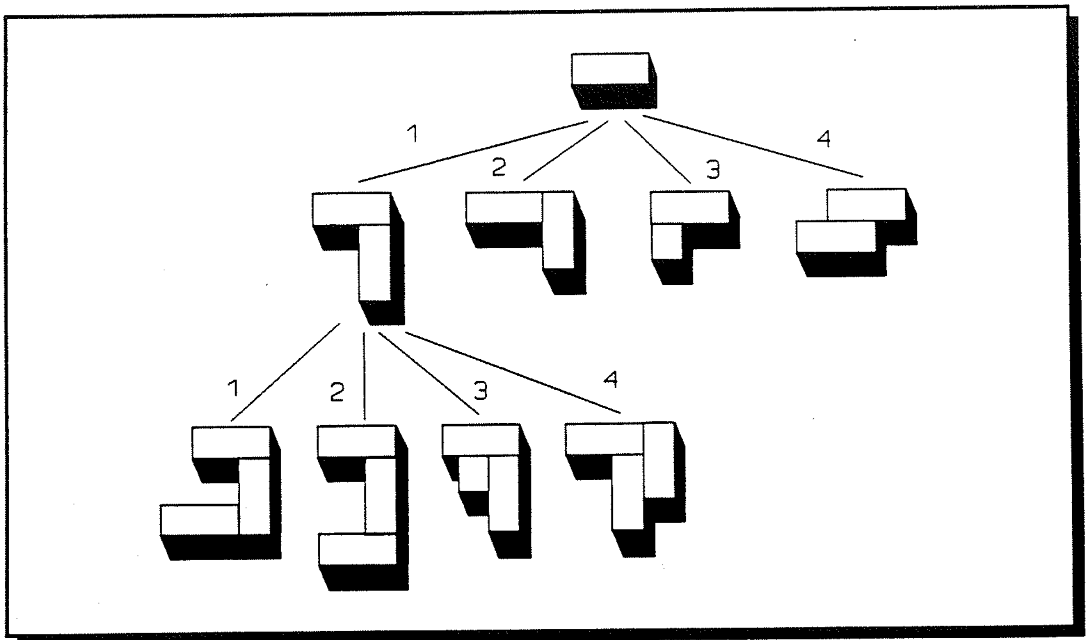


Figure 3-9: Layout alternatives for a three-space configuration based on the rules in figure 3-8

(March and Steadman 1971). Based on these results, several research efforts were undertaken to expand the transformation concepts into a sound design methodology. These efforts culminated in the ground breaking work of Stiny, which expanded the formalism to the currently accepted shape grammar definition (Stiny 1980).

The shape grammar formalism revolves around the concept of shape rules. Each rule defines a set of configuration conditions and an action which specifies how an additional shape may be added to an existing configuration. A given set of these shape rules represents a specific shape grammar, or designer vocabulary. Figures 3-8 and 3-9 illustrate how the shape grammar paradigm generates configurations, given the four shape rules in figure 3-8. Each configuration in the bottom row of figure 3-9 represents a legal configuration based on the defined shape grammar.

At the outset of shape grammar research, the emphasis remained on the definition of grammars for well-known architects. These efforts resulted in defined grammars for several architects including Frank Lloyd Wright and Palladio (Stiny 1985). However, the development of knowledge-based system paradigms, and the close relationship between shape rules and expert system production rules, provided an impetus for researchers to implement the shape grammar formalism in an expert system format. Within these efforts, researchers investigated the potential of combining additional overall design goals with the basic shape rule concept (Gero and Coyne 1985b), examined the underlying concepts of design (Stiny 1990), and created new formalisms for design theories based on original shape grammar representations (Flemming 1978).

Advantages

The shape grammar formalism symbolizes a bridge between algorithmic-based systems and knowledge-based systems. Shape grammars formalize geometric manipulation strategies into design languages, while capturing design knowledge related to the placement of spaces during configuration generation. The combination of these two approaches addresses the expandable theory issue and the design knowledge issue. Additionally, this combination distinguished shape grammars from algorithmic-based formalisms, since it addressed a non-adjacency based requirement. In contrast to the adjacency concerns of algorithmic-based systems, shape grammars focus on the overall concerns of layout rationale and form. This addresses one of the primary criticisms of algorithmic-based systems and presented the first approach to generating layouts with an underlying rationale.

Disadvantages

Although shape grammars diverge from an adjacency requirement reliance, they retain a single attribute emphasis. Placement options are exclusively generated according to the design concept captured within the shape grammar rules. Thus, the remaining design influences are excluded from the placement generation process.

Based on this limitation, the issue arises as to whether shape grammar research should be pursued in its pure form, or in combination with additional reasoning paradigms. The former approach is being pursued based on the argument that the capability to generate form through defined grammars is in itself a useful design goal (Chase 1989; Knight 1989). The latter approach addresses the multiple attribute limitation through the addition of layout analysis systems which evaluate the configurations in relation to design program requirements (Flemming et al. 1988). The selection of a particular approach depends on whether the research objective is to explore pattern generation, or the potential to generate and evaluate configurations according to design program requirements.

Layout Analysis Systems

Layout analysis research transfers the focus of knowledge-based systems from layout generation to the analysis of previously created layouts. However, the rule-based reasoning paradigm evident in the previous categories remains the fundamental reasoning paradigm. Current research efforts are applying knowledge-based paradigms to the analysis of several factors impacting the design of given spaces or buildings. Preliminary applications include acoustical analysis (Pohl et al. 1988), daylighting analysis (Pohl et al. 1988), fire code compliance (Dym et al. 1988), and ventilation analysis (Pascall and Hamilton 1990).

The emergence of these research efforts is attributable to the available bodies of information detailing the processes required to perform analysis functions. This information provides researchers with the ability to develop analytical systems by translating accepted procedures into knowledge-based formalisms. For example, in the fire code domain, researchers translated existing fire code regulations into layout analysis rules. These rules were then included within an overall framework emulating accepted procedures for detecting code violations (Dym et al. 1988).

Advantages

Layout analysis systems provide the potential to check configurations for attribute compliance based on expertise captured within a knowledge-based system. Therefore, the multiple attribute and external influence limitations associated with several layout generation methodologies including shape grammars and expert systems, could be overcome through post-processing by layout analysis systems. Layout analysis further assists system development by permitting incremental component development. Rather than requiring all attributes to be addressed from the start of development, individual layout analysis systems can be incorporated as the opportunity exists. This incremental development is permitted based on the separation of analysis tasks. Developing each analysis task separately as a self-contained component which does not require access to additional analysis systems, permits these tasks to be incrementally added to an overall system with minimal impact on the remaining components. Based on the spectrum of complex issues influencing layout configurations, this incremental development is necessary to address the multiple interrelationships between design attributes.

Timing of Layout Analysis

Layout analysis research has resulted in the emergence of a basic research question, "When should layout analysis occur?" The examples presented in this section illustrate analysis occurring at the conclusion of layout generation. The advantage of this approach is that all information required to perform an analysis task is evident in the final layout. For example, an egress compliance analysis requires all corridors to be in place for a definitive layout analysis to occur. Additionally, analysis results may be placed in an overall layout context, rather than a single layout segment. In certain instances, such as the determination of energy consumption, the overall layout is the preferred perspective from which to evaluate analysis results.

However, some analysis tasks may be more appropriately performed during the layout generation process. Specifically, eliminating infeasible placement options based on design program requirements is required as individual spaces are added to the layout. The analysis of these options as they are generated permits the system to either accept or reject the options at the time of analysis. If the option is unacceptable, then the system eliminates the time required to pursue the partial configuration. Conversely, if the option is acceptable, the system may focus its attention on this option with the assumption that it represents a viable partial solution.

The second strategy is incorporated within the CAADIE approach. The project is based on the assumption that a single configuration can be generated through an aggregation of individually accepted placement options. Achieving this objective requires the elimination of unacceptable options at the time they are created. This elimination can only occur if analysis is performed throughout the layout generation process.

Integrated Systems

The final category of systems within this review combine generation and analysis systems from various design disciplines into integrated design environments. Although the CAADIE research effort does not encompass design discipline integration, the intent to incorporate various layout generation domains in one system exhibits a relationship to integrated system research in terms of cooperative problem solving concepts. Furthermore, the lack of previous work in the development of cooperative layout generation environments necessitates this broader design integration review.

The evolution from independent components to integrated environments is prompted by the interactions that occur between design discipline experts. For example, during conceptual design, interaction occurs between architects and structural engineers to determine appropriate structural systems. Interactions such as this continue throughout the design process, and increase as the number of conflicting design concerns increase in the final design stages. Due to these interactions, several inter-disciplinary research projects have been undertaken to investigate the integrated design environment issue (Pohl et al. 1989; Fenves et al. 1989; Sriram et al. 1990). Although each effort addresses this issue through a different strategy, each contains a cooperative problem solving concept.

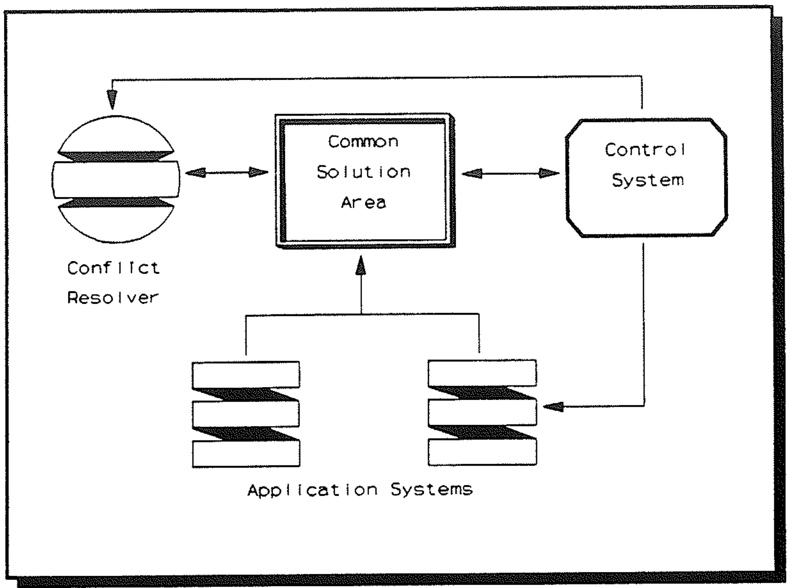


Figure 3-10: The integrated problem solving environment

Figure 3-10 presents a conceptual view of an integrated design environment. In this figure, the application systems represent knowledge-based or algorithmic-based systems focusing on specific design problem concerns. Each system places and retrieves design information in the common solution area. This solution area provides access to the current solution state. In some instances, systems may independently generate partial solutions which impact partial solutions generated by other application systems. In these circumstances, the conflict resolver retains the responsibility to resolve any design attribute conflicts. Finally, the control system incorporates the algorithms and knowledge required to control the overall operation of the environment. The combination of these components creates the cooperative problem solving premise evident in the current integrated system research efforts.

The recent introduction of this research emphasis inhibits the inference of any conclusive findings. However, the preliminary studies and implementation tests demonstrate the importance of several cooperative problem solving concepts:

- The development of a flexible control structure to permit the pursuit of any design sequence

- The development of a negotiation strategy to determine which design constraints should be relaxed when a conflict arises between equally important design participants
- The development of an underlying structure to ensure continued designer participation throughout the design process (i.e., the system must not operate as a black box designer)

Current Research Directions

The previous sections summarize three decades of layout generation research approaches and methodologies. The overview is intended to serve as an introduction to these techniques. Further reviews of several efforts have recently been published by Rychener (Rychener 1988), Coyne (Coyne et al. 1990), and Schmitt (Schmitt 1988a). The reader is directed to these reviews for further, in-depth analyses of the previous methodologies.

The following sections diverge from this overview to expand upon several current research efforts addressing issues, or incorporating goals, that are related to the CAADIE project. For each project, the review presents an overview and an analysis to spotlight the research implications. Specific instances are noted where CAADIE research builds upon work presented in these research efforts.

The LOOS Research

The LOOS project is currently being undertaken at Carnegie-Mellon University under the direction of Professor Ulrich Flemming. The project is based on research previously conducted on the use of shape grammars to generate layouts containing loosely packed rectangles (Flemming 1986). In this research, Flemming formalized an approach for generating placement options incorporating hollow areas between spaces. These loosely packed rectangles address layout domains such as kitchen design and furniture layout, which require objects to be placed in a distributed manner within the layout boundary. Based on this work, the LOOS effort focuses on incorporating these geometric theorems into a system which addresses broader layout generation concerns. The LOOS objectives are as follows:

- The ability to systematically enumerate alternative solutions to a design problem
- The ability to take, at the same time, a broad spectrum of design criteria or concerns into account (Flemming et al. 1988, 1)

The principal concept used to implement these objectives is a generate-and-test philosophy. The system architecture includes an additive component which uses a set of shape rules to generate all possible configurations for each stage in the layout evolution. The testing component then evaluates each configuration for compliance with a given set of user provided criteria, and a set of domain specific design rules. The resulting configurations are used to generate the next stage of layout options. The researchers consider the generation and testing of all potential alternatives as an opportunity to evaluate the complete range of options available at any given time. This enhances the ability of the system to examine trade-offs between alternatives in terms of meeting conflicting design requirements (Flemming et al. 1988, 77).

Analysis

The LOOS project achieves prominence based on its ability to address multiple layout generation issues while retaining a mathematical basis. In contrast to design methodology approaches, shape grammars provide the capability to justify all placement options based on geometric theorems. Thus, the subjectivity of the placement generation process is reduced. Furthermore, the system fragility in terms of addressing multiple domains is diminished due to the use of domain independent grammars, rather than domain specific generation rules. The concept oriented grammars focus on architectural forms rather than design requirement conditions. Thus, in most domains, the same set of shape rules can be invoked to generate form-based configurations. In contrast, domain dependent rules must be altered to address domain specific conditions. The distinction of this domain independent element is illustrated by the application of LOOS to diverse problems including kitchen design and high-rise building core design.

Disadvantages

The drawback of the LOOS system returns to its reliance on shape grammar formalisms. The shape rules guiding the generation of placement options are based on the synthesis of form. The rules do not incorporate the remaining range of attributes influencing layout generation. Rather, the rules generate a fixed set of options under all circumstances. The analysis systems determine the compliance level of the options **after** all possible placement options are generated. Although the exhaustive enumeration of possibilities will inevitably produce options complying with the requirements, the static nature of the rules requires exhaustive enumeration at every point in the process. The rules do not incorporate either sufficient knowledge, nor integrated evaluation procedures, to intelligently adapt to changing design circumstances. It is this adaptation capability which separates systems consolidating decision making knowledge within the placement generation process, from shape grammar based systems.

ABLOOS

A second research endeavor undertaken in conjunction with the LOOS project is an effort to extend the problem solving strategy incorporated within the LOOS system (Coyne and Flemming 1990). Whereas, LOOS addresses every object in a layout generation problem at a single level of abstraction, Abstraction-based LOOS (ABLOOS) addresses objects at various levels of abstraction. Therefore, rather than attempting to individually place every object into the overall layout, ABLOOS permits designers to decompose the layout task into several individual subtasks. For example, a problem could have the goal of placing several groups of furniture in a large room. Within ABLOOS, this goal can be decomposed into several subtasks including the placement of the required areas for each individual furniture group, and the placement of individual furniture pieces within each furniture group. This capability is intended to support problems with a greater level of complexity by enabling the system to break the problems down into manageable components. Thus, although each component can be solved as an independent problem, the individual solutions combine to address the overall goal for the given layout problem.

CAADIE Influence

The adaptation issue influences all knowledge-based layout generation systems. The CAADIE system confronts design adaptation through a combination of design heuristics and requirement analysis functions. In contrast to the exhaustive enumeration of options, this approach emphasizes the generation of options in accordance with current layout requirements, design concepts, and appropriate design principles. In this manner, a limited number of options are explored based on designer preferences and design principles to obtain the desired option. This approach is intended to provide the adaptation capability lacking in the shape grammar formalism.

The Charrette Research

The Charrette project has been undertaken at the US Army Corps of Engineers' Construction Engineering Research Laboratory (CERL) to develop a system which assists the designer in the conceptual stage of design. The system is based on a research study conducted to determine the requirements for a successful

CAAD system (Bond et al. 1988, 11-13). Based on the completion of this study, the following objectives were outlined for the Charrette research project:

- Include enough flexibility to accommodate personal variation in design methods
- Be transparent to the user
- Use profession-specific language
- Accept and retrieve preliminary building requirements and criteria
- Accumulate decisions and help the designer use past experience to improve current and future designs
- Allow flexibility in these decisions
- Take advantage of the user's knowledge wherever possible

(Bond et al. 1988, 12-13)

The Charrette implementation of these objectives centers on three conceptual design activities identified as possible design process elements. These activities include developing spatial requirements for spaces, generating bubble diagrams, and generating block diagrams from the previous bubble diagrams. The system is intended to facilitate these actions through an interactive format consisting of a series of menu options and a series of activity specific modules.

Analysis

The issue separating the Charrette research from many concurrent efforts is the multiple design stage emphasis. In contrast to many layout generation systems, this effort identifies several conceptual design stages in which to evolve the layout alternative. This emphasis is based on studies conducted by the Charrette researchers identifying the conceptual design phase as the most applicable environment for CAAD assistance. Subsequently, the studies established that the development of bubble diagrams from layout requirements, and the development of block diagrams from bubble diagrams, are two areas which are prevalent in the conceptual design process (Bond et al. 1988, 26-39). At the present time, the Charrette system represents one of the few systems to specify these areas as a potential layout generation framework.

Based on the initial state of the Charrette research, it is inappropriate to analyze implementation issues. However, one system deficiency is evident in the initial research objectives, the lack of an evaluation component. As discussed in relation to the design process issue, an iterative process incorporating analysis, synthesis, and evaluation components is essential in supporting architectural design. The lack of an evaluation component eliminates the potential to provide compliance feedback in accordance with specified criteria. In the absence of this feedback, the system loses an important interactive element. Specifically, the designer loses the capability to use the system as a platform for studying the ramifications of altering design program requirements.

CAADIE Influence

In response to this deficiency, the CAADIE approach incorporates evaluation as an integral layout generation process stage. The evaluation component provides designers with layout ratings based on design program compliance. Given this feedback, designers may pursue various issues and options by comparing the evaluation ratings given for each layout alternative. Thus, the evaluation component contributes to the participatory design environment and iterative design process concepts identified for the CAADIE project.

Designer Expertise Research

The research being conducted by Omer Akin at Carnegie-Mellon University was addressed previously in reference to the HeGeL layout generation system. However, this section emphasizes the continuing work by Akin to study the reasoning performed by architects during layout conceptualization (Akin, Dave, and Pithavadian

1988; Akin 1988). In accordance with the emphasis on introducing conceptual design tools, Akin spotlights the importance of capturing this area of design expertise in the following passage:

> The single phase of the complex process in which the architect is still the sole decision maker is that of preliminary design. It is generally believed that the essentials of the architect's creation are shaped during this phase (Akin 1988, 176).

Based on studies of problem solving methodologies used by architects, Akin identifies four significant expertise areas within the conceptual design process: scenarios, alternatives, evaluation, and prototypes (Akin 1988, 180-181). Scenarios represent an architect's use of overall organizational ideas for a given design solution. Alternatives represent the architect's use of different design constraints to perform "what-if" type explorations. Evaluation includes the expertise permitting architects to evaluate both partial and overall designs. Finally, prototype expertise includes the knowledge a designer utilizes from previous design solutions to assist in generating initial concepts for the current design circumstance.

The identification of these design expertise areas has contributed to the CAADIE knowledge model development effort by providing one of the first documented recommendations for incorporating specific design knowledge areas within a knowledge-based CAD system. Although the CAADIE research findings refine these categories in accordance with the layout generation problem, these studies provide a valuable reference point for the CAADIE research effort.

Analysis

In contrast to Flemming, Akin proposes emulating designer actions in relation to how layout generation is performed including how placement options are generated, and how problems are structured to emphasize particular attributes. The assumption in this approach associates the amount of system knowledge with the capability to generate acceptable layout configurations. If sufficient knowledge can be captured, then the system will be capable of addressing and structuring layout problems in the same manner as experienced designers.

This approach diverges from Flemming's reliance on exhaustive enumeration, by relying exclusively on captured knowledge to guide the generation of placement options. This knowledge determines what spaces to select, which attributes to address, how to generate options, and how to restructure layout problems. Thus, the strength of the computer to generate options based on geometric algorithms and layout requirements is bypassed in favor of design heuristics. This limits the range of potential options considered, to those locations identified by the design heuristics. Given the studies by Simon on the limitation of human problem solving capabilities (Simon 1981), the question arises as to whether or not the captured knowledge will contain similar problem solving limitations. Specifically, the range of attributes addressed at any given time, and the number of options considered, will be limited to the cognitive capabilities of the design experts.

CAADIE Influence

As discussed previously, the CAADIE approach emphasizes a combination of designer and computer strengths. In relation to Akin's research, the knowledge model includes design knowledge related to placement decisions such as space selection and conflict resolution. However, the approach diverges from Akin by retaining the geometric manipulation and compliance evaluation functions within algorithmic procedures. As will be discussed later, this approach permits the design knowledge to generate a structured context in which the geometric algorithms generate placement options.

The ICADS Research

The ICADS research project is currently being undertaken at the CAD Research Unit at California Polytechnic State University, San Luis Obispo. This research diverges from the previous projects due to its integrated system emphasis. Given this emphasis, the following review highlights the resulting impact on the CAADIE project, rather than focusing on the system's design capabilities. The ICADS project focuses on the development of an intelligent, computer-based design environment to support architects and engineers during the design process. This effort encompasses three major component areas:

> - The development of a CAD database management system to store both graphic and non-graphic design attributes
> - The development of an expert design advisor to monitor the design process and provide expert design advice
> - A multi-media presentation facility for various graphic capabilities during the evolution of design
>
> (Pohl et al. 1988, 10-11)

These objectives are addressed in the first ICADS project prototype, ICADS Working Model Version I. However, the relevance of this research to the CAADIE project precedes the prototype implementation studies. In developing this prototype, the researchers closely examined the multiple attributes affecting the design of spaces and buildings (Pohl et al. 1988). Based on this study, a taxonomy of design attributes was developed to categorize the attributes and the design information components related to each category. To illustrate, within the lighting attributes category, a sub-category of daylighting incorporates the design information component stating the percentage of lighting which should be fulfilled by natural light. This categorization of attributes and identification of information components provided a strong basis for the initial CAADIE knowledge model development.

The second relevant area emphasizes the user control issue. The ICADS researchers studied potential alternatives for designer participation including video, evaluation feedback, and drawing tools. Based on these studies, the potential for flexible user control options became quite evident. Subsequently, the CAADIE project identified the implementation of a participatory design environment as a principal system objective. Through this objective, it is intended that the designer will retain the prerogative to control as much, or as little, of the layout generation process as is appropriate for the current circumstances. Concurrently, the system will adjust its role to provide the appropriate level of design assistance.

Summary

The numerous layout generation research efforts provide a substantial base of information and approaches to the problem. These efforts cover a wide spectrum of computational efforts from traditional mathematical approaches such as the quadratic assignment efforts, to current knowledge-based approaches such as the use of expert systems. As illustrated in figure 3-11, these approaches achieve varying levels of success in terms of addressing the layout generation problem issues. An analysis of these efforts provides several key points for consideration within future layout generation systems.

- Algorithmic-based methods do not provide, in their base formalisms, sufficient capabilities to capture the reasoning processes followed by designers during the conceptual design process.
- The capture of conceptual design expertise represents the greatest opportunity for impacting the overall layout design.
- The development of flexible user control options enhances and encourages designer participation, and ultimately, will increase the potential for designer acceptance.
- The integration of several design domains is required to address the multiple issues associated with layout generation.

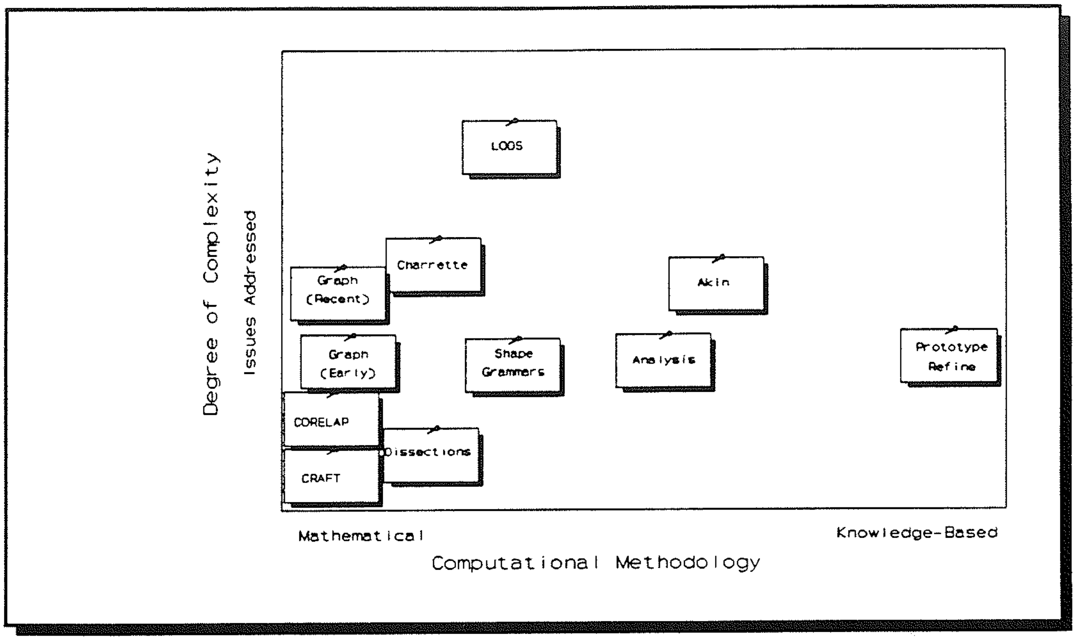


Figure 3-11: A comparison of the varying degrees of success each computational methodology has achieved in addressing layout generation issues

In terms of the CAADIE research effort, the previous approaches serve to establish several detailed objectives for the CAADIE prototype implementation:

- The capture of design knowledge to guide the **decision processes** within the overall layout generation process.
- The development of algorithmic procedures to generate placement options within the guidelines imposed by the design program and design heuristics.
- The use of an iterative analysis-synthesis-evaluation cycle as the central layout generation process.
- The incorporation of layout analysis functions within the placement generation process to reduce the number of infeasible options addressed by the system.
- The development of a framework to capture both the layout information components and layout heuristics used in the layout generation process.

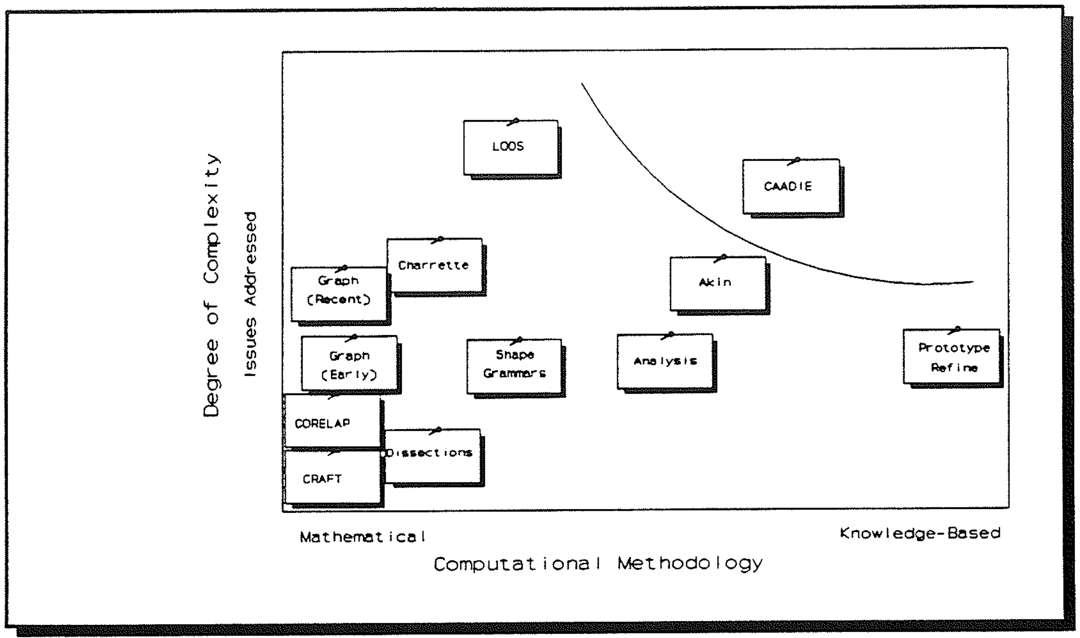


Figure 3-12: The relationship between the CAADIE research efforts and the previous layout generation research efforts

- The establishment of criteria on which evaluation feedback can be provided to assist in the exploration of layout alternatives.
- The development of a flexible control structure to permit various components, each of which focuses on a specific layout generation issue, to interact throughout the layout generation process.
- The design of a user interface which permits designers to exercise as much, or as little, control over the generation process.

These emphasis areas place the CAADIE research effort in an evolutionary position along the graph illustrated in figure 3-12. Through the use of established knowledge-based paradigms, the CAADIE effort extends the number of layout generation issues addressed by concurrent knowledge-based attempts, and overcomes several limitations restricting algorithmic-based methodologies from addressing the true complexity of the layout generation problem.

CHAPTER 4

THE CAADIE APPROACH

Each of the layout generation approaches reviewed in the previous chapter encompasses a unique set of objectives from which to address the layout generation problem. Subsequently, the objectives guided the implementation of prototype systems to investigate the viability of the layout generation approach. As previously illustrated in figure 3-12, the success of these systems in addressing these issues fluctuates with the individual methodologies. However, as each approach has evolved with successive research efforts, the resulting systems have addressed a greater number of layout generation issues. This potential to address a greater number of issues through the evolution of layout generation approaches forms the basis of the CAADIE approach.

> **CAADIE Approach:** Address an expanded scope of design and computational issues by utilizing knowledge-based paradigms to capture design knowledge and design principles which guide the generation of layout configurations within the framework of a participatory design environment.

The following sections, and next three chapters, introduce the implementation strategies selected to develop a prototype system based on this approach. This chapter presents an overview of the system through discussions of the knowledge acquisition process and the CAADIE system architecture. Supporting this overview in the following chapters are reviews of the knowledge model, the layout generation process, and the user interface.

Knowledge Acquisition

The knowledge acquisition process traditionally represents the primary bottleneck in knowledge-based system development (Buchanan et al. 1983). This bottleneck appears due to the extensive amount of interview time required to extract and capture desired knowledge from domain experts. The interview process requires several iterations of obtaining and analyzing knowledge to ensure that all relevant domain knowledge is identified. In the architectural design domain, the acquisition process is further complicated by the subjective nature of the domain and the habitual manner of working developed by individual designers (Magee 1987).

In contrast to domains such as automobile repair or medical diagnosis, architectural design is not judged in terms of right or wrong solutions. Rather, a design is judged in terms of subjective issues such as its application of design principles (Ching 1979). Furthermore, this judgement is dependent on the designer's interpretation of the design program requirements. These subjective elements permit multiple design solutions to exist which may each be viable alternatives. Selecting a "correct" answer from these alternatives is subject to the preferences of the designer. Thus, the objective of knowledge acquisition in the design domain diverges from capturing knowledge which derives correct answers, to capturing knowledge focusing on the fundamental application of design knowledge and principles.

A further knowledge acquisition process complication arises from the difficulty of extracting knowledge related to primary decision making processes such as the selection of spaces for placement, the resolution of design conflicts, and the selection of appropriate placement options. The repeated use of this knowledge throughout each design problem results in the knowledge becoming an ingrained part of the designer (Minsky 1986). Thus, the designer no longer explicitly recollects using this knowledge, rather, the knowledge is used as an intuitive design process component. Subsequently, designers have a difficult time verbalizing this knowledge during knowledge acquisition sessions. This requires, as discussed below, an approach other than designer interviews to extract the knowledge required for design systems to incorporate knowledge-based decision making processes.

Knowledge Acquisition Process

The knowledge acquisition process employed during the CAADIE development phase incorporated two primary stages: the selection of multiple design sources and the use of design scenario studies.

Design Source Selection

The multiplicity of issues associated with the layout generation problem results in designers acquiring different perspectives on, and preferences related to, the layout generation process. However, at the center of each perspective, lies a core of established decision making guidelines and design principles, which are used to guide layout generation. It is this core of design guidelines and principles which formed the focus of the knowledge acquisition process. To ensure that this common core of design decision making capabilities could be identified and captured, several designers were selected with varying backgrounds and perspectives.

The group of designers selected for the knowledge acquisition process represent a limited cross-section of the architectural design profession. However, for a proof of concept study, this cross-section provided a representative group from which to extract a segment of the required core knowledge. The group included a practicing architect, a design educator, and two design consultants. It should be noted, that this group was not selected for their expertise in any given area, or with the assumption that their knowledge was complete and representative of the entire architectural design profession. Rather, the group was selected for their diversity and willingness to participate in the CAADIE project.

Design Scenarios

Several studies on the acquisition of design knowledge demonstrate that the most effective method of obtaining this knowledge is through protocol studies (Akin 1988; Magee 1987; Akin 1979). In these studies, a designer is given design scenarios to analyze and solve. As the problem solving session proceeds, the designer verbalizes the reasoning and decisions leading to a problem solution. Thus, the designers are prompted into bringing forth the ingrained knowledge used to generate design decisions. Through this acquisition process, the underlying knowledge required to develop knowledge-based design systems is made available to design researchers.

The CAADIE protocol studies employed the use of design scenarios to extract specific areas of design knowledge. For example, the following scenario segment illustrates the type of scenarios used to acquire conflict resolution knowledge.

> ...Given the lack of available placement options due to the listed design attribute conflicts, analyze the current configuration of spaces, and the associated layout requirements, to determine which requirements should be relaxed to resolve the conflict.

When presented with the scenarios, the designers provided considerable knowledge related to several areas including their design experience, the use of design principles, and the influence of design attribute requirements. The responses were then compared to separate common design knowledge from individual preferences and experiences. Once this separation was completed, a further separation occurred to extract knowledge focusing on decision processes such as selecting spaces for placement. This high-level decision making knowledge represents the reasoning strengths identified as essential to the CAADIE approach of capturing reasoning and information processing strengths within a layout generation system.

Upon completing the knowledge separation process, several iterations occurred where each interviewee was presented with the core knowledge extracted during the design scenario sessions. This verified that each designer was in agreement with the determination of which knowledge could be considered as a common basis. Subsequently, the knowledge was categorized into specific emphasis areas, and used as a basis for the prototype system development.

System Architecture

The CAADIE prototype required the implementation of a participatory design environment incorporating designer knowledge, design principles, and layout information. Figure 4-1 illustrates the resulting system architecture. The architecture emphasizes a modular implementation, with four components comprising the system foundation: the user interface, the knowledge model, the layout generation knowledge sources, and the controller.

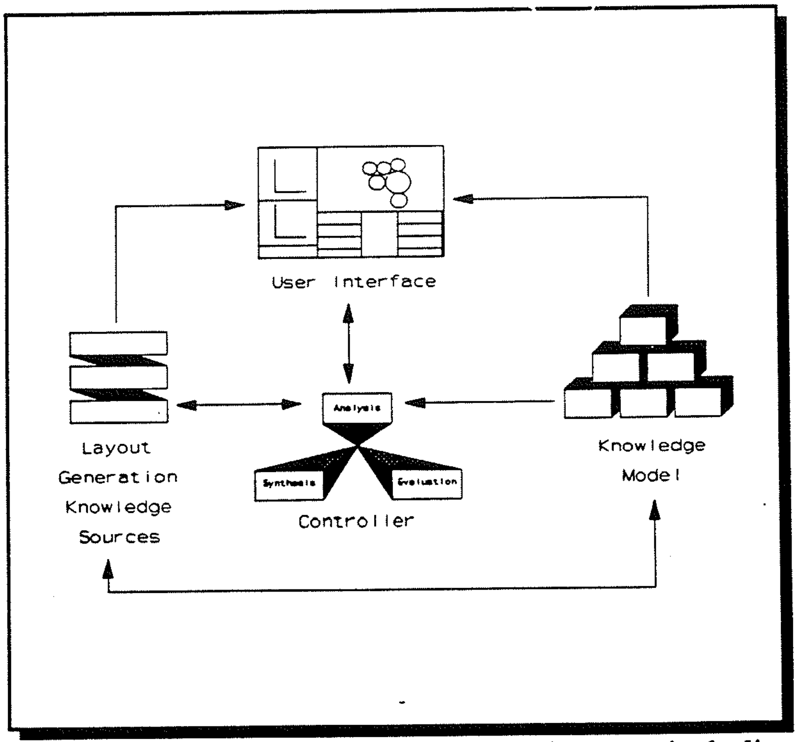


Figure 4-1: The CAADIE system architecture including system components and information flows

The individual components are implemented in LISP on the KEE™ platform developed by IntelliCorp, Inc. This platform was selected based on a combination of formal and informal justifications. The formal justifications are based on the knowledge representation factors defined by Duce and Ringland for selecting reasoning platforms (Duce and Ringland 1988). Specifically, the expressive capability to support multiple reasoning and representation paradigms, and the built-in reasoning and representation primitives were the primary formal considerations for selecting the KEE platform. In terms of the informal justifications, the availability of the platform, and the potential portability of the KEE knowledge bases were primary considerations. The prototype currently runs on a SUN workstation.

The following sections provide an overview of the CAADIE architecture components. The user interface, knowledge model, and layout generation knowledge sources are briefly introduced as a prelude to the extended discussions in the following chapters. The remaining component, the controller, is discussed in detail at this point based on its role as the central element through which all system actions are initiated.

User Interface

The principal function of the user interface is to create a participatory design environment in which a designer cooperates with the system during the layout generation process. The interface provides graphic facilities for designers to impart and receive layout information throughout the generation process. Input including design preferences and user requests, are entered through interface menus and selection boxes. Similarly, controller messages and knowledge source solutions are displayed to the designer through interface windows and dialogue boxes. The user interface provides these capabilities by building upon the multi-window capabilities within the KEE platform. The multiple windows create a participatory environment by enabling various types of graphic information including layout relationship diagrams, evaluation charts, and interactive requirement indicators, to be simultaneously displayed during the layout generation process.

Knowledge Model

Whereas the user interface provides the designer with the necessary tools to participate in the layout generation process, the knowledge model provides the CAADIE system with the necessary knowledge and information to perform its role in the participatory design process. The knowledge model supports this role by forming a repository for the layout generation information and knowledge required to perform system initiated actions. The system components access the knowledge model to obtain typical guidelines and requirements for the current layout problem. Similarly, the components access the model to obtain the knowledge required to address these requirements in the context of the current layout generation instance. Chapter 5 introduces the organization of this information and knowledge, together with proposed representation formats.

Layout Generation Knowledge Sources

The layout generation knowledge sources perform the tasks required to generate layout configurations. Knowledge sources transform the layout information and knowledge stored in the knowledge model into layout solutions. Within the design context, the knowledge sources represent design consultants, each of whom has access to a general body of knowledge, but specializes in addressing a single knowledge area. Analogously, the knowledge sources themselves access the knowledge model, however, each focuses on a different segment of the knowledge to perform specific design tasks. This specialization permits the set of CAADIE knowledge sources to be altered with minimum impact on the remaining components. Since each knowledge source performs its given task independently of the other knowledge sources, these knowledge sources may be altered, replaced, or additional ones added to the system, without interfering with other layout generation tasks.

SLOTS	FUNCTION
Action	The design task performed by the knowledge source is defined within this slot. The actions do not require any activation parameters other than a message to the slot name.
Trigger	The trigger slot contains the conditions under which the knowledge source is eligible to perform its task.
Worth	The worth slot defines the relative value of the knowledge source compared to the remaining eligible knowledge sources under the current circumstances.
Focus	The design stage or task which the knowledge source addresses
Sub Focus	The specific area of knowledge source influence, e.g., internal or external attributes
Score	The final calculated score during controller evaluation

Figure 4-2: The slots defining each CAADIE knowledge source

Knowledge source implementation is based on a common structure comprised of the slots illustrated in figure 4-2. The slots define applicability conditions, relative importance, and layout generation actions. These definitions represent the baseline requirements for including a knowledge source within the system, and provide the avenues through which the controller interacts with the individual knowledge sources.

System Controller

In contrast to the predetermined, step-by-step procedures defining tasks such as assembling furniture pieces, layout generation requires designers to adapt to evolving layout circumstances. Specifically, designers must alter their decision making processes to address specific design influences as the circumstances warrant. For example, if a space requires external access, then the designer must address this requirement at both the time the space is placed, and whenever another space potentially affects this access. Thus, the underlying requirement is for the designer to remain flexible during the design process to address issues as the circumstance requires, and the opportunity exists.

Approaches for achieving this flexibility within a computer-aided design system are predominantly based on variations of an opportunistic control strategy initially introduced in the speech recognition domain (Erman et al. 1980). In this strategy, a controller activates knowledge sources based on their applicability to the current problem requirements. By comparing the current circumstance to knowledge source applicability, the controller activates the knowledge sources required to adapt the system focus to the problem requirements. Thus, the system strategy focuses on reacting to opportunities presented during the solution generation process.

The opportunistic control strategy has been adapted to address the particular requirements of individual problems including protein analysis (Hayes-Roth 1984), sonar interpretation (Nii 1986), and medical supervision (Hayes-Roth 1988).

However, within the computer-aided design domain, the predominant variations on the opportunistic control strategy center on either a central flow of control approach (Fenves et al. 1989), or a cooperative control approach (Pohl et al. 1989).

Central flow of control. The distinctive feature within the *central flow of control* strategy is the sequential problem solving process. Within this approach, the controller iterates between determining knowledge source applicability, and activating a selected knowledge source to perform its respective design task. At the completion of each iteration, a segment of the design problem has been addressed by the knowledge source with the greatest applicability. Through this iterative process, only one activity takes place at any given time. This approach contains the advantage of retaining a consistent problem solving methodology both throughout the design process, and for every variation of knowledge sources. The iterative process ensures that the flow of control always returns to the controller at the completion of a design task. This provides the capability of isolating knowledge source activation responsibility within a single system element.

The principal disadvantage of this approach is the inability to provide a concurrent monitoring capability. The interrelationships among design attributes often result in several requirements being affected by an individual design change. For example, the reduction in length of an exterior wall could, at a minimum, influence the lighting, structural, and HVAC requirements of the spaces adjoining the wall. In a design meeting, consultants in each of these areas could immediately notify the designer of this impact. However, the *central flow of control* approach restricts this capability by allowing only a single active consultant (i.e., knowledge source). Thus, a minimum of three iterations would be required to analyze the lighting, structural, and HVAC consequences resulting from the exterior wall alteration.

Cooperative control. The *cooperative control* approach alleviates the inherent *central flow of control* disadvantage by permitting multiple knowledge sources to be concurrently active. In this strategy, the controller emphasis focuses predominantly on conflict resolution rather than knowledge source activation. Since several knowledge sources are monitoring the design process at a single time, the controller focuses on resolving conflicts between knowledge source recommendations. In this sense, the controller serves as a designer mediating disputes among pairs of design consultants. The advantage of this approach is the capability to simultaneously monitor a greater number of design attribute requirements. Thus, if a designer alters a configuration, then immediate feedback is available from all affected knowledge sources. Similarly, if one knowledge source proposes altering the configuration, then the remaining knowledge sources can evaluate the alteration without controller activation delays.

The primary disadvantage of this approach is the difficulty in regulating the proportion of generated information. The automatic analysis by multiple knowledge sources can result in several simultaneous recommendations. This condition may require extensive controller analysis to determine the relevance of the generated recommendations. For example, given the previous exterior wall adjustment, knowledge sources may either generate designer warnings, generate layout requirement alternatives, or generate new spatial configuration alternatives. Given this proliferation of design changes, the controller is required to analyze the changes for design conflicts. Potentially, this generation of information could be extensive enough to result in system delays.

A second disadvantage of this approach is the multi-processing requirement. In contrast to the sequential actions in the previous approach, the *cooperative control* approach requires multiple knowledge source activation throughout the problem solving process. To support this capability, the system requires a multi-processing environment. This specific type of environment restricts the range of software platforms which may be used to implement a cooperative control strategy.

CAADIE Implementation

The relative advantages and disadvantages of each control strategy provide a basis for justifying the selection of either strategy for system control. However, based on the CAADIE prototype requirements, the *central flow of control* strategy has been selected for the CAADIE prototype. The final selection included the following primary considerations:

- Predictable flow of control - The assurance that the controller regains control after each design task, provides greater monitoring of knowledge source interactions during prototype development.
- Domain independent controller - The reduced analysis requirements within the *central flow of control* approach provides the capability to implement the controller with domain independent knowledge. The controller may thus be applied to a greater range of domains with fewer alterations to the underlying selection knowledge (Hayes-Roth 1985).

The Control Process

The CAADIE control implementation is modelled after the approach introduced by Hayes-Roth (Hayes-Roth 1984). In this approach, the controller iterates through a three-step cycle: knowledge source selection, knowledge source ranking, and knowledge source execution. In each cycle, a knowledge source is activated to perform a specified task. When the task is completed, the controller regains control to determine the next knowledge source task. The following sections, together with figure 4-3, detail the process as it is implemented in the CAADIE prototype.

Selection

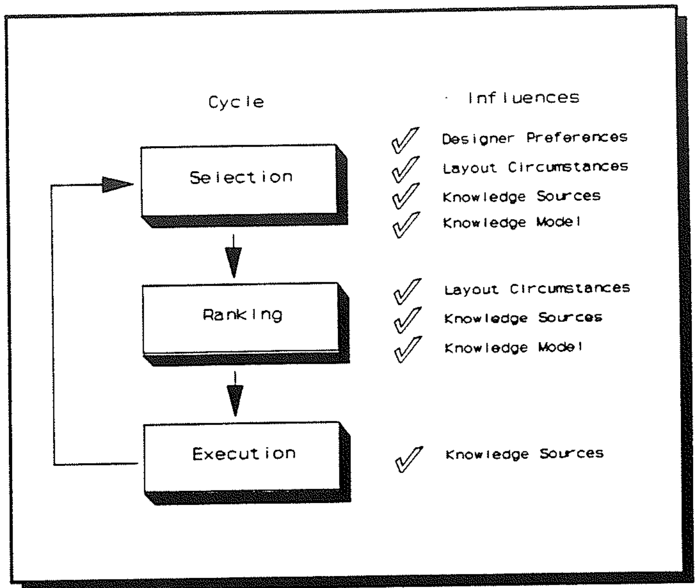


Figure 4-3: The CAADIE control cycle

The selection stage determines the applicability of each knowledge source to the current layout circumstance. This determination is based on the applicability condition captured in the *trigger* slot of each knowledge source. This condition may be either a reference to a previously completed task such as an analysis task, or a series of conditions related to several layout circumstances including the current participatory generation mode and the user interface requirements. The controller compares these applicability conditions with the current state of the layout generation process. If the current circumstances fulfill the conditions of any knowledge source, then the knowledge source is considered eligible for activation.

Ranking

Based on design attribute interrelationships, several knowledge sources will be eligible for activation at the completion of certain design tasks. However, due to sequential processing restrictions, only one of these knowledge sources may be activated in each control cycle. Thus, a method of ranking is required to determine which knowledge source has the greatest relative benefit to the current design process state.

The CAADIE controller achieves this ranking through a two-step process. In the first stage, the controller evaluates the *worth* slot of each eligible knowledge source. This slot may either contain a numeric value representing a constant knowledge source ranking, or a LISP function which calculates the knowledge source ranking based on the design circumstance and the remaining eligible knowledge sources. These options provide the flexibility to create rating mechanisms commensurate with the requirements of each knowledge source. Thus, while a knowledge source focusing on initial layout analysis may always contain a high rating, a knowledge source related to evaluation concerns may contain a function giving it a lower priority while analysis knowledge sources are eligible, but a higher priority when other knowledge sources are available.

The *worth* slot evaluation may result in a highest rated knowledge source. However, due to attribute interrelationships, several knowledge sources could contain equal ratings. In this circumstance, the controller incorporates design heuristics as a second stage in guiding knowledge source selection (figure 4-4). These heuristics determine final knowledge source rankings based on emphasis areas, previous actions,

IF	The focus of the previous knowledge source was synthesis
AND	The focus of the eligible knowledge source is evaluation
AND	The subfocus of the eligible knowledge source is external
AND	The score of the eligible knowledge source is equal to the high score
THEN	Increase the score of the eligible knowledge source by 1

Figure 4-4: A knowledge selection heuristic emphasizing external attribute evaluation

and attribute priorities. Therefore, the final knowledge source selection is based on designer preferences rather than algorithmic-based methodologies.

Activation

The final control cycle stage activates the design task within the *action* slot of the highest rated knowledge source. Based on the object-oriented paradigm of message passing, the controller activates the selected knowledge source by sending a generic message to the *action* slot. This message subsequently prompts the knowledge source to perform its defined task. Thus, controller independence is facilitated by eliminating the need to provide the controller with specific requirements for activating each knowledge source.

Knowledge source activation results in the performance of a specific layout generation task. The completion of this task subsequently results in a design state alteration. Given this alteration, a new set of conditions and emphasis areas emerge as guidelines for the next design task. To perform this task, the flow of control returns to the controller for the purpose of selecting the next applicable knowledge source.

Summary

The CAADIE system architecture is a collection of modular components, each of which is designed to support a segment of the layout generation process. The architecture features include:

- Participatory design environment support.
- Component independence to permit system alterations.
- Support of a knowledge-based approach to layout generation.

- Provision for multiple design emphasis areas through multiple knowledge sources.
- An opportunistic strategy for controlling layout generation based on adaptation to evolving problem conditions.

Given these features, the architecture provides the basis for addressing the project objectives within the CAADIE prototype. The following chapters introduce the specific components within the prototype which support layout information and knowledge representation, layout generation, and designer participation.

CHAPTER 5

THE KNOWLEDGE MODEL

The fundamental objective of the CAADIE knowledge model is to capture the categories of layout generation knowledge and information used by designers. While previous work by researchers including Stiny (Stiny 1980) and Gero (Gero and Coyne 1985a), has explored capturing segments of this information and knowledge, requirements for representing the broad spectrum of knowledge affiliated with the layout generation process has received less attention. The following model introduces one potential framework for this representation.

The model stresses the premise that the consolidation of both layout generation knowledge and layout information is essential to the successful generation of knowledge-based layouts. If a system is fixated strictly on layout generation knowledge, then it reduces the capability to use layout information as constraints and guidelines for the evolving layout. The lack of these constraints and guidelines renders the eventual solution inferior in terms of the actual layout requirements. Conversely, if the system is fixated entirely on satisfying design constraints, then the system loses the capability to generate solutions based on designer experience and intuition. This form of design produces layouts which may satisfy the specified design constraints, however, the layouts lack the underlying organization provided by the application of design principles.

Knowledge Categories

The layout information categories identified in the CAADIE study include: topological attributes, design attributes, and spatial ordering concepts. Concurrently, the layout generation knowledge categories include: designer expertise, design attribute heuristics, spatial ordering heuristics, and knowledge selection heuristics.

Layout Information Categories

The classification of layout information into topological attributes, design attributes, and spatial ordering concepts, represents the framework within which layout information is captured in the model. The following descriptions summarize the types of information found in each category.

<u>Topological Attributes</u>. The layout information represented in this category provides the topological constraints and guidelines for individual design objects such as spaces or floors. These attributes include typical square footage requirements, typical dimensions, typical length-width ratios, and other information related to the physical dimensions of the design object. These design attributes are isolated in the system based on indications from designers that topological attributes are often addressed separately during the design process. Topological attributes provide designers with elementary information related to the feasibility of placing spaces in particular locations. For example, the guidelines will indicate if a location will accommodate the area requirements of a particular space.

<u>Design Attributes</u>. This category represents a consolidation of two design issues, multiple design attributes and external influences. The information included within this category includes typical requirements for each attribute such as acoustic levels, daylighting provisions, and access requirements. These issues are combined due to their common trait of influencing the design of individual objects. Throughout the layout generation process, a designer will disregard the external versus internal

focus of these issues and address the issues interchangeably. For example, although privacy and view considerations emerge from different design issue categories, they are collectively addressed as attributes influencing particular design objects during placement generation. In the absence of specific preferences, both attributes may be treated equally in terms of compliance determination.

<u>Spatial Ordering Concepts</u>. Spatial ordering concepts provide the underlying basis for the evolution of layout configurations. Specifically, these concepts represent the decisive rationale for layout generation decisions. To support the selection of these concepts, this category contains information related to the applicability of concepts to particular building types and specific design attributes. For example, linear concepts are more applicable than clustered concepts for layouts emphasizing security, due to the potential for isolating high security areas from other parts of the building. Similarly, clustered concepts are more appropriate for elementary school campuses than other building types due to educational and social considerations.

Layout Generation Knowledge Categories

The categories of designer expertise, design attribute heuristics, spatial ordering heuristics, and knowledge selection heuristics, provide the framework for layout generation knowledge within the model. The following descriptions summarize the types of knowledge found in each category.

<u>Design Attribute Heuristics</u>. This category comprises the knowledge a designer utilizes to incorporate design attribute information such as daylighting provisions and space planning requirements, into the layout generation process. This knowledge is structured in accordance with the layout generation stages to facilitate the use of the same design attribute information at different times. For example, during analysis, attribute knowledge is required to generate layout requirements from available space planning information. However, during synthesis, additional

knowledge is required to use the same information to determine spatial placement option preferences. Similarly, specific heuristics are required for each of the remaining design attributes to focus on the intrinsic characteristics of each attribute. For example, addressing access requirements necessitates a notably different area of knowledge than is required to address daylighting guidelines. Consequently, the design attribute heuristics category comprises knowledge related to the application of design attribute information in each layout generation phase.

<u>Spatial Ordering Heuristics</u>. The generation of spatial placement options will differ according to the selected spatial ordering concept. For example, the placement options generated for a clustered layout will vary from those generated for a linear layout. Furthermore, for each concept, design principles influence the selection of preferred placement options. For example, given two linear concept options, the one reinforcing the linear form to the greatest extent will be the preferred placement option. Thus, heuristics in this category generate placement options according to design principles and selected spatial ordering concepts.

<u>Knowledge Selection Heuristics</u>. The knowledge selection heuristics represent the knowledge a designer utilizes to determine when to alternate between various phases of the design process and when to emphasize design expertise areas. In terms of the design process, this knowledge serves to determine when analysis of constraints should be completed, when synthesis of spatial options should commence, when evaluation of options should be invoked, and finally, when these phases should iterate to refine the evolving layout. Concurrently, this knowledge assists the designer in analyzing the conditions under which specific types of design knowledge should be selected to assist in the layout generation process. For example, given the choice of activating an externally or internally focused knowledge source, the knowledge selection heuristics determine the appropriate selection according to previous knowledge source selection, current preferences, and knowledge source applicability.

Designer Expertise. Designers bring to each layout generation problem a level of expertise developed from addressing similar problems over a period of time. Through this experience, designers develop heuristics and preferences for addressing several layout generation issues such as resolving attribute conflicts, selecting spaces for placement, and transitioning between layout generation stages. These heuristics guide designers in making decisions which impact the overall configuration throughout the layout generation process. To increase the decision making capabilities of the knowledge base, this category captures knowledge to perform high-level activities such as conflict resolution and space selection.

Knowledge Model Representation

The representation of layout information and knowledge is accomplished through frame hierarchies and rule sets, respectively. These representation paradigms provide the flexibility necessary to incorporate the wide spectrum of issues identified as pertinent to addressing the true complexity of the layout generation problem. Additionally, as discussed further in the following sections, these representation paradigms provide the flexibility to focus layout generation decisions at the level of individual design objects. For example, daylighting information from the design attributes category is represented in a manner permitting daylighting related decisions to be made at the individual space level. The following descriptions summarize the requirements for, and use of, each representation paradigm.

Why Represent Layout Information?

The primary reason for representing layout information is to reduce the amount of information a designer is required to provide. In early layout generation systems, layout information was excluded in favor of an approach which required designers to provide all layout requirements (Grant 1983a). Criticisms were levelled at this approach by design professionals based on a perceived inconsistency between the assistance provided and the effort required to input the layout requirements (Chase 1990). However, the narrow spectrum of attributes addressed by these systems made it possible to employ this approach.

Subsequent layout generation research diverged from this approach based on two principal reasons: increased attribute requirements and prototype identification. In the former case, the increased number of attributes addressed by layout generation systems began to stretch the limits of designer reliance. It became infeasible to rely on designers to continually input the ever increasing amount of layout information. Concurrently, as discussed previously, design and CAD researchers were putting forth the argument that a notable segment of the design process entails adopting typical requirements and guidelines to current design problems (Alexander and Poyner 1970; Jones 1979). Through these two circumstances, research began to move towards storing typical design requirements to reduce input requirements, and subsequently, designer reliance.

This argument prompted development of the information representation facilities within the CAADIE knowledge model. Based on the information category definitions, the knowledge model incorporates typical requirement values for each design attribute defined in the prototype. The accumulation of these values reduces reliance on designer input by providing the framework for the automatic generation of requirements. Therefore, designers may emphasize design activities rather than information input activities.

Frame Hierarchies

The layout generation process requires designers to address a diversity of design attributes according to the current design phase. Each of these attributes contains specific requirements and guidelines which impact the layout generation process. However, these attributes are not independent pieces of information. Rather, the designer views these attributes as components within the design object context to which they belong. In architectural terms, these design objects are typically defined as Neighborhood, Site, Building, Floor, Space, Workstation, Wall, and Window (Pohl et al. 1989). For example, the daylighting requirements of a space are not addressed as an independent design program requirement. Rather, the designer addresses this requirement as one attribute guiding the design of the overall space. Thus, a designer requires the information related to the specific design object to be contained within the context of that object.

The CAADIE knowledge model facilitates this requirement to address design information in relation to design objects through the use of frame hierarchies. Frame representations permit various attributes to be defined in the context of a single data element. To illustrate in an architectural context, window attributes such as size, clarity, and opening type, may be aggregated within a single frame referred to as "window". This capability to combine information into a single frame context, parallels the requirement to address multiple design attributes in a single design object context. In addition, the inheritance capability of frame structures permits the hierarchies to be described as a series of class specializations (Fikes and Kehler, 1985). Within this specialization process, objects at lower levels of the hierarchy are characterized by their greater specificity in terms of attributes and values. For example, faculty offices may be described as offices plus a set of properties and attribute values that distinguish faculty offices from other kinds of offices. Thus, the frame hierarchies provide the capability to define and allocate both general and specific design attribute information according to the requirements of each design object.

Topological Hierarchy

The CAADIE knowledge model defines two primary frame hierarchies for depicting layout information: the topological attributes hierarchy and the design attributes hierarchy. The topological hierarchy defines the fundamental design object organization of the knowledge model (figure 5-1). This organization is based on the architectural design objects comprising layout configurations, and is explicitly defined at the *design object focus* level. At this level, each class represents a generalized definition of the required design objects. Subsequently, the hierarchy specializes into subclasses defining the distinctive design objects found in particular building types. This specialization proceeds until, at the lowest level of the frame hierarchy, the complete set of design objects found in the building type are represented by individual subclasses.

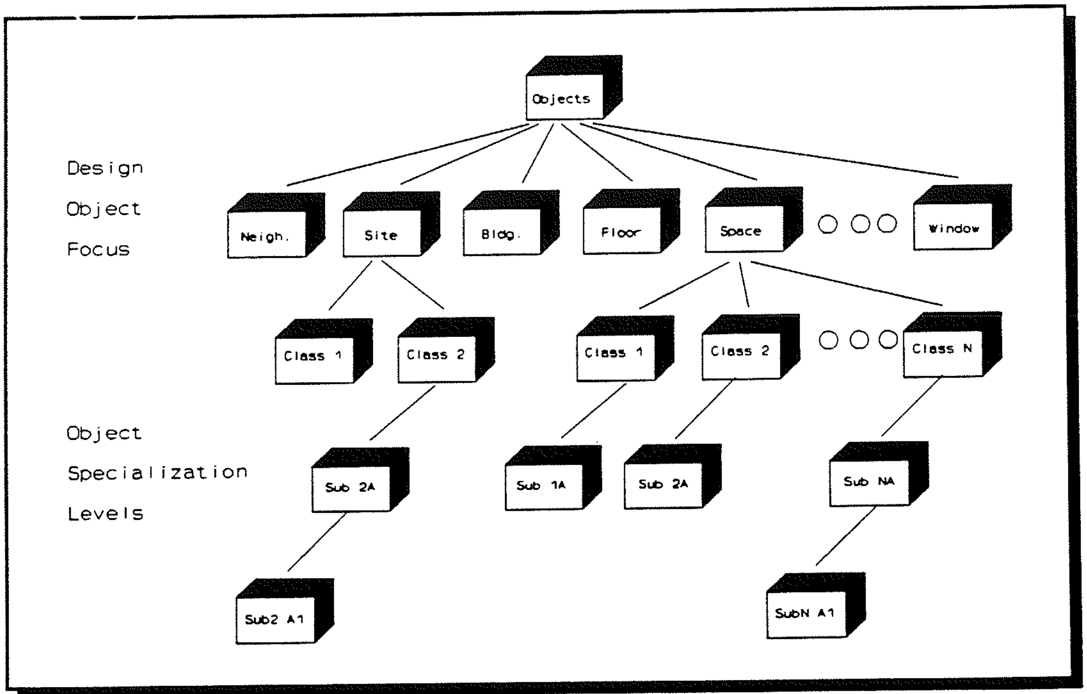


Figure 5-1: A general model of the topological hierarchy

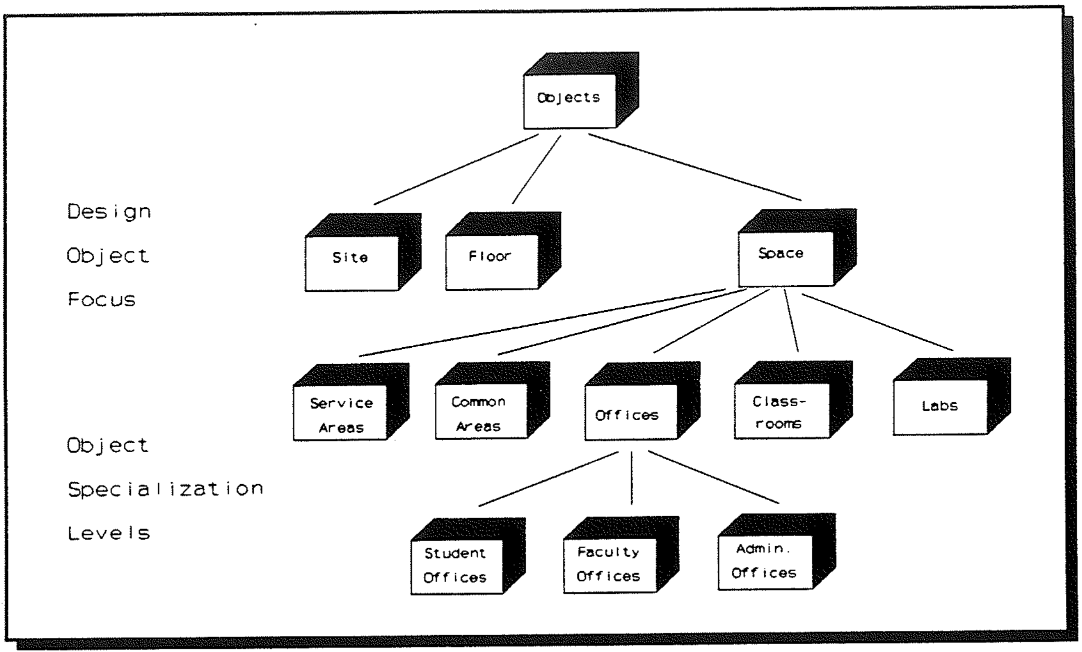


Figure 5-2: The instantiated topological hierarchy

Within the CAADIE system, the topological hierarchy is instantiated with design objects related to the system domain of university research buildings. Figure 5-2 illustrates a segment of this hierarchy highlighting the specialization from a generic *spaces* class to a specific *faculty offices* subclass. As the hierarchy progresses from generic classes to specific subclasses, the definitions become increasingly tailored to faculty office requirements. This specialization is repeated in the definition of the remaining space types such as classrooms and open areas. In each of these class-subclass definitions, the focus remains on providing the flexibility and support to transfer layout information to the design object context.

Within the topological hierarchy frames, typical design object attributes are stored at the appropriate level of specificity (figures 5-3 and 5-4). For example, default office dimensions are stored in the *offices* frame, and specific faculty office dimensions are stored in the *faculty offices* frame. In this way, if specific information is not provided in the current design stage, default office dimension information may

SLOTS	FUNCTION
Area	The square footage required for the space.
Length	The typical length of the space
Width	The typical width of the space
Variance	The percent variance allowed to go over the typical area. This permits earlier stages to account for the less refined space dimensions.
X Coordinate	The x coordinate of the space within the current layout.

Figure 5-3: Topological attributes defined in the spaces frame

SLOTS	FUNCTION
Max EW	The east-west distance of the buildable area
Max NS	The north-south distance of the buildable area
Current NS	The current north-south distance occupied by the partial configuration

Figure 5-4: Topological attributes defined in the floors frame

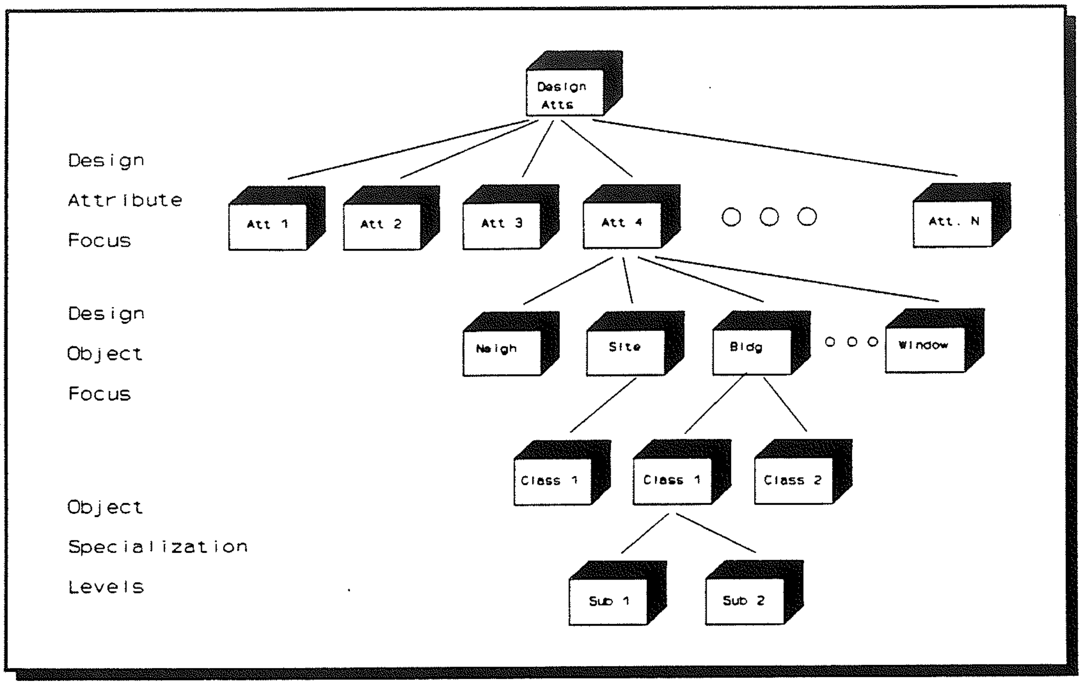


Figure 5-5: A general model of the design attributes hierarchy

be inherited down through the hierarchy to the *faculty offices* frame. However, if specific faculty office information is obtained, then it will override the general *offices* information.

Design Attributes Hierarchy

The design attributes hierarchy complements the topological hierarchy by defining the framework for including design attribute information. However, the notable variation between the two models is the initial level of model organization. Whereas the topological hierarchy focuses entirely on a single attribute type, the design attributes hierarchy must transfer various types of attribute information to the appropriate design objects. Figure 5-5 illustrates the CAADIE proposed definition for the attributes hierarchy. Two pertinent points concerning the hierarchy definition require further discussion.

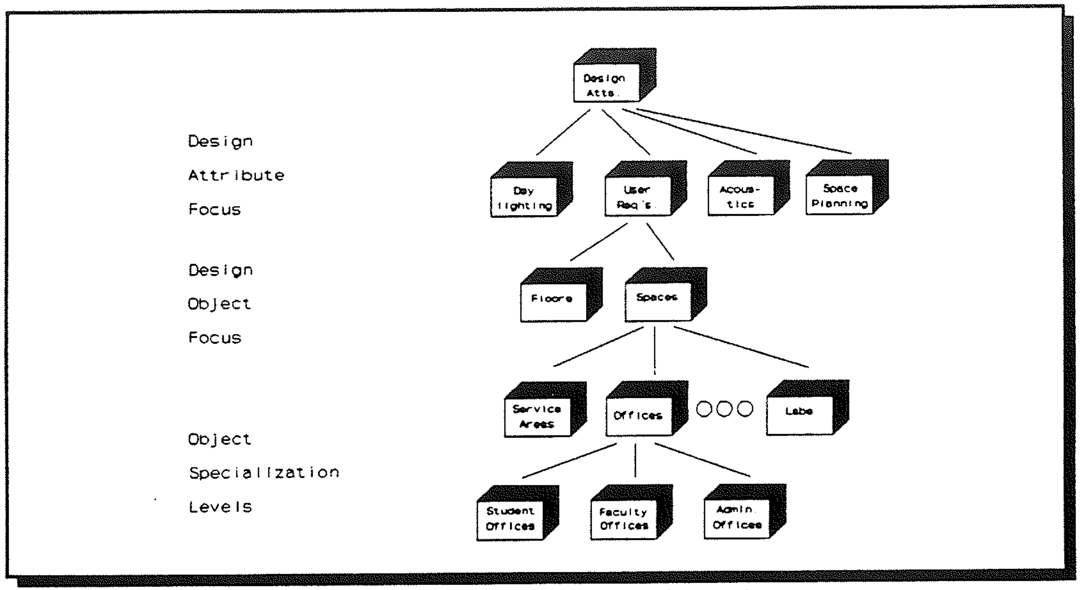


Figure 5-6: The instantiated design attributes hierarchy

- Initial definitions focus upon design attributes - To support the designer practice of organizing design attribute information in relation to individual attributes, the initial hierarchy level is organized according to these attributes. This separation provides the organizational structure in the hierarchy. Through this structure, the addition and updating of design attribute information is isolated to focus upon specific design program areas. In contrast to the haphazard collection of information which results from combining all attributes in a single frame structure, the design attribute organization separates attribute requirements into specific information categories. In this way, information related to an individual attribute is isolated, and may be updated independently from the remaining design attributes.
- Transfer design attribute information to design objects - The initial design attribute classes revolve around design attribute categories. However, to comply with the overall model focus on design objects, the design attribute

information must be transferred to the context of individual design objects. To accommodate this transfer, the third level of the hierarchy, the *design object focus* level, emulates the design object hierarchy defined in the topological hierarchy. The design objects specialize into subclass definitions representing the same set of design objects included at the lowest level of the topological hierarchy. Thus, at this lowest level, design attribute information is organized into subclasses defined in terms of both design attributes and design objects.

Figure 5-6 illustrates a segment of the instantiated design attributes hierarchy focusing on user requirements. At the *design attribute definition* level, the user requirement slots common to all design objects are specified. Subsequently, user requirements are defined for each design object until subclasses such as the *faculty offices* subclass, contain specific user requirement information (figure 5-7). Thus, the hierarchy facilitates both structured information organization, and an overall design object emphasis.

Design Attribute Representation

The design attributes incorporated in the knowledge model hierarchies encompass numerous attribute guidelines and requirements. Representing these guidelines and requirements involves the use of value representations and design object relationships to accommodate the individual attribute requirements.

Value Representations. Value representations encompass both numeric attribute values and symbolic attribute values. Numeric attribute values represent typical topological attributes such as square footage, lengths, and widths. In contrast, symbolic attribute values are required to support attributes such as daylighting, which contain values not amenable to simple numeric representations. These values focus

SLOTS	FUNCTION
Acoustic Level	The typical noise levels allowed by and generated by the space. For example low/med translates to low noise tolerance and medium noise output
Privacy Level	The typical privacy level required for the space (i.e., public, semi-private, or private)
Security Level	The typical security level required for the space (i.e., high, med, or low)
View Type	The typical view requirement for the space. This value does not include the specific direction. Rather it indicates primary or secondary views.
View Direction	The specific view direction determined by the analysis knowledge sources.

Figure 5-7: The user requirements slots defined for each space

on abstract concepts such as compass directions and daylighting types, which support attribute related issues such as design object orientation and placement. Additionally, these values are instrumental in using design knowledge centered on non-numeric issues such as appropriate view allocation.

Design Object Relationships. The second representation method focuses on representing relationships between design objects. These relationships occur as either subclass relationships, or as individual object relationships. Subclass relationships enable the hierarchy to transfer information related to an entire group of spaces to a single design object. For example, a positive adjacency relationship contained within the *faculty offices* subclass indicates that each faculty office has a positive relationship to all other faculty offices. Similarly, individual object relationships permit information related to particular design objects to be retrieved by other objects containing relationships to that object. For example, to check for security violations between two spaces, the security requirements of each space may be retrieved through the relationship link stored in the adjacency slot of the respective spaces (Figure 5-8). Thus, substantial amounts of information from other spaces is made available to an individual space, while eliminating the necessity to explicitly represent the information.

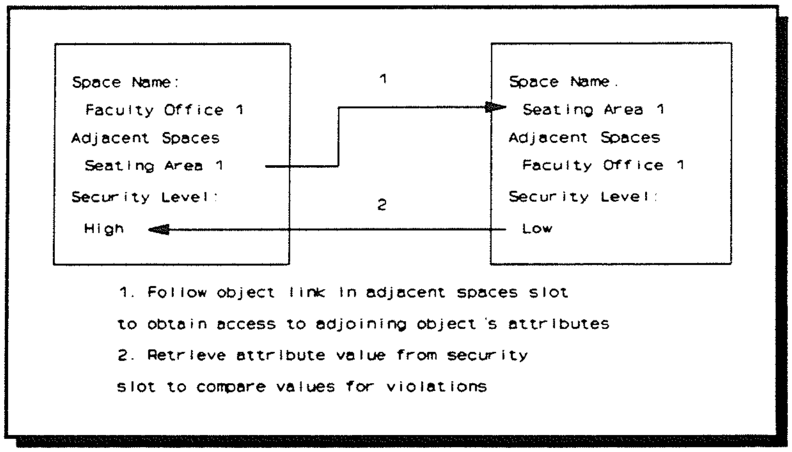


Figure 5-8: Design object references permit information to be retrieved from frame instances

Rule Sets

The CAADIE rule set representation features the heuristic knowledge associated with the layout generation knowledge categories. For each knowledge category, the preferences and principles associated with the category are translated into corresponding IF-THEN representations. The selection of this format is based on the flexibility provided by the generic IF-THEN paradigm. Within this format, knowledge related to any of the four knowledge categories may be represented with a common representational format. The rules may capture either designer heuristics, designer preferences, or design principles. Furthermore, the rules contain the flexibility to address layout information captured within the frame hierarchies. For example, the following rule illustrates a simple design heuristic to select a placement option based on adjacencies and daylighting requirements.

```
IF    Space 1 contains a positive adjacency with space 2
AND   Space 2 contains the same daylighting requirements as Space 1
AND   Space 2 has an available side to place space 1
THEN
      Place space 1 adjacent to space 2
```

Although this is a simple rule, it illustrates the potential to access layout information from individual rules. In the CAADIE system, where various types of information are accessed by different layout generation knowledge categories, easy access to this information becomes imperative to support all stages of the layout generation process. The combination of frame access and english-based representation make rules an ideal format for this task.

Rule Set Modularity

The CAADIE knowledge model incorporates rule sets to combine rules into logical collections. These rule sets consolidate related heuristics into coordinated groups which can be activated at appropriate layout generation stages. For example, knowledge source selection heuristics are grouped within a rule set which is activated when the system requires knowledge source selection.

The benefit of this rule set organization lies in the modularity provided to the CAADIE knowledge base. A documented restriction of knowledge-based systems is the difficulty of updating and altering system heuristics (Jackson 1986, 218). Rule-based systems become unwieldy to update due to complex interrelationships, redundancies, and conflicts. These problems are exacerbated in design systems when individual design preferences are considered. In contrast to rule-based domains where documented procedures exist, architectural design is dependent on the experience of the individual designer. To support these individual experiences, the rule sets need to be replaceable with a minimum of implications to the remaining knowledge base. The modular rule sets support this requirement by separating heuristics into specific focus areas. The limited focus of the rule sets provides the ability to replace specific knowledge areas with new rule sets containing updated or alternative designer preferences.

The next chapter outlines the impact and use of these rule sets, and the frame hierarchies, in the context of the layout generation process.

CHAPTER 6

LAYOUT GENERATION PROCESS

The layout generation process in the CAADIE prototype employs an iterative analysis-synthesis-evaluation cycle as the underlying problem solving framework. Within this iterative cycle, the system alters between layout requirement generation (analysis), spatial configuration generation (synthesis), and spatial configuration evaluation (evaluation).

Analysis

The analysis stage emphasizes the acquisition and generation of general and space-specific layout requirements. General requirements comprise guidelines and constraints impacting the overall layout including design attribute importance and preferred spatial ordering concepts. Space-specific requirements incorporate typical guidelines impacting the design of each space including security levels, sunlight exposures, and noise levels.

General Layout Requirements

The acquisition of general layout requirements stresses designer-system interaction. This interactive approach is required due to the experience which designers bring forth to each layout problem. This experience influences designers to select certain preferences during design program development such as the selection of design concepts, focus attributes, and the importance of design attributes. These preferences adapt knowledge and experience to the current problem. Thus, it is imperative to capture these preferences as a method of adapting the system to a designer's set of desired parameters, and implicitly, to reflect the designer's experience and intuition.

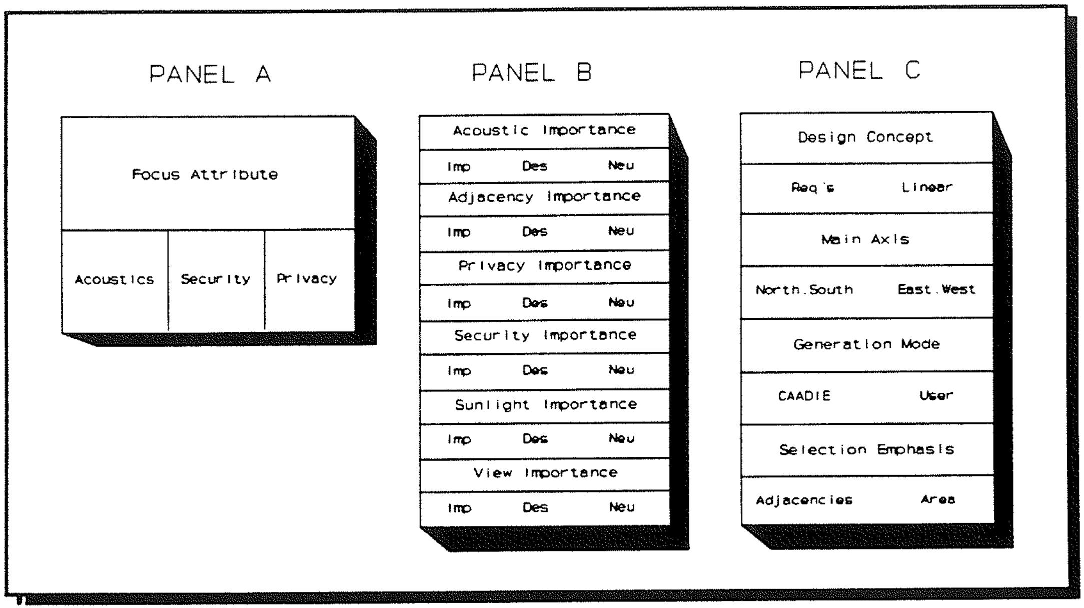


Figure 6-1: The input panels for general layout requirements

Designers impart general layout preferences by interacting with the selection panels illustrated in figure 6-1. PANEL A provides a choice of design attributes for the underlying layout organization. Based on this selection, internal zones are created to emphasize the focus attribute requirements (i.e., public vs. private, high security vs. low security, and high noise tolerance vs. low noise tolerance). In the absence of conflicting attributes, spaces with similar focus attribute values are placed together in appropriate zones.

In PANEL B, the designer sets the global importance factors for each design attribute. These importance factors determine which attributes should receive greater compliance emphasis during layout generation. For example, in a project emphasizing natural light, the designer may determine that the importance of complying with daylighting constraints is greater than that of complying with noise requirements. Although these preferences take precedence in the majority of cases, specific spaces will override these preferences based on stored requirements. This procedure permits spaces to identify specific attributes as being either more or less important. For example, in the previous example, classrooms will override the overall reduction in noise importance based on established requirements for complying with specific noise levels.

PANEL C provides the designer with preferences related to the overall form of the layout. The design concept option prescribes whether the form should evolve from the layout requirements, or emphasize specific characteristics. If the linear concept is preferred, then the main axis option provides alternate directions for establishing the axis. The selection preference supports these concepts at a space-specific level. After all other preferences have been considered, this option selects spaces for placement based on either size or number of adjacency constraints. Finally, the generation option defines the generation mode for the synthesis stage. The designer may either request the system to generate layouts based on design program requirements, or elect to personally generate layout solutions.

Finally, the designer provides the buildable area in which the layout can be generated. These maximum north-south and east-west dimensions can influence the configuration by forcing the layout into an elongated shape. For example, if the north-south dimension is very small in relation to the east-west dimension, then the flexibility to expand the configuration in the north-south direction will be limited. Thus, the final solution will be elongated along the east-west direction to comply with the north-south restriction. In contrast, if the buildable area is large enough to accommodate any configuration of spaces, then the buildable area will not impact the final solution.

The designer posts the buildable area dimensions directly to the frame representing the floor on which the spaces will be placed. In a future implementation, these values may be included within an input panel. However, at this time, it is undetermined if specific design object values should be intermixed with general layout preferences.

Space-Specific Requirements

The generation of space-specific requirements diverges from the interactive approach by focusing on the utilization of knowledge model information. Whereas general requirements are inherently particular to a given circumstance and individual designer experiences, individual space requirements tend to remain constant for particular building types. Therefore, to reduce user input requirements, CAADIE utilizes typical layout information in the knowledge model, combined with analysis knowledge sources and rule sets, to generate space-specific requirements. The system achieves this objective through the four-step process illustrated in figure 6-2.

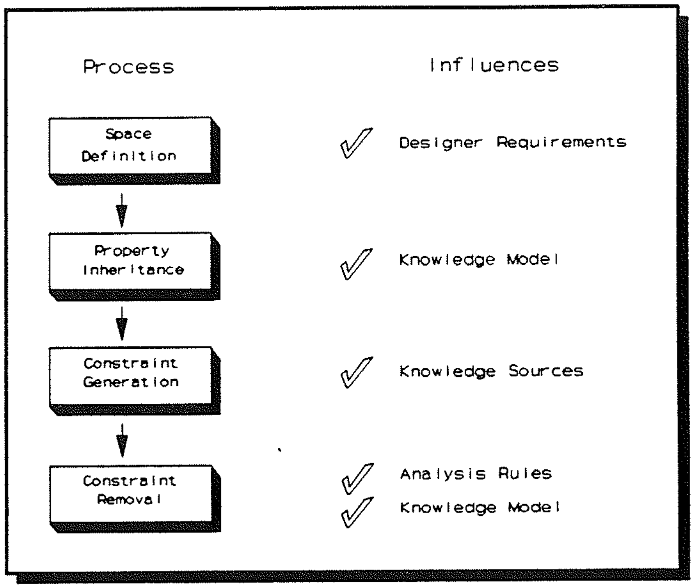


Figure 6-2: The generation process for space-specific requirements

Space Definition

The space definition step initializes the system by defining the spaces required for the current layout. This step constitutes the system boundary in terms of the initial design process phase. The CAADIE prototype assumes that a preexisting design program defines the required spaces. Given this assumption, CAADIE relinquishes the responsibility of defining spaces to the designer.

Spaces are defined in an ASCII file containing a predefined format based on KEE syntax requirements (Appendix A). This file is loaded into the system when the designer specifies a filename during the initialization process. The file defines the units required for representing spaces in the knowledge model. It is anticipated that this definition process will eventually be replaced by a front-end design program module.

Property Inheritance

Upon loading the space definition file, CAADIE generates a complete set of space-specific requirements based on information stored in the frame hierarchies. The hierarchy descriptions emphasize a focus on design entities such as faculty offices and classrooms, as the final level of subclass definitions. The effectiveness of this definition strategy becomes apparent during the property inheritance phase. Specifically, multiple inheritance provides the capability to transfer space-specific requirements from design object subclasses to individual space instances.

Figure 6-3 illustrates the generation of requirements for an individual faculty office. In this example, a segment of the two knowledge model hierarchies are depicted with their respective attributes defined in the *faculty office* subclass. As the space definition file is loaded into the system, and a faculty office requirement is acknowledged, a *faculty office* frame is created as an instance of **each** *faculty office* subclass defined in the attribute hierarchies. This instantiation permits the frame to inherit the attributes defined in each *faculty offices* subclass. Similarly, each of the remaining user defined spaces will inherit a set of space-specific requirements by becoming instances of the respective frame hierarchy subclasses.

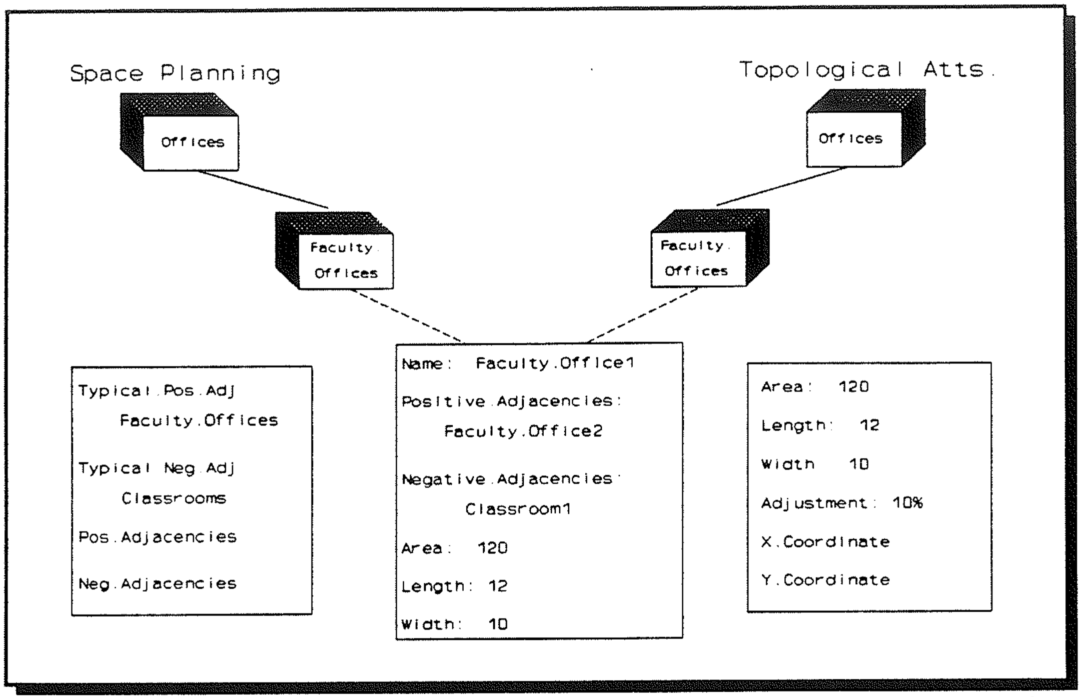


Figure 6-3: The generation of space-specific requirements through multiple inheritance

Specific Constraint Generation

The requirements and guidelines inherited from the attribute hierarchies are predominantly values which are directly applicable during the layout generation process. However, several requirements such as views and adjacencies, require an additional analysis phase to generate specific layout requirements. The common trait among these requirements is the dependence on layout circumstances which cannot be foreseen prior to the actual layout generation circumstance. An analysis knowledge source addresses this issue by generating specific requirements during the initial analysis stages.

The generation of individual view requirements necessitates specific knowledge of the views existing on each site. In response to this individuality, the knowledge model retains the *type* of view required for each subclass of spaces (i.e., primary, secondary, any, or none). The analysis knowledge source examines these values, and the site's primary and secondary views, to appropriately transfer the directional values to the individual spaces (figure 6-4).

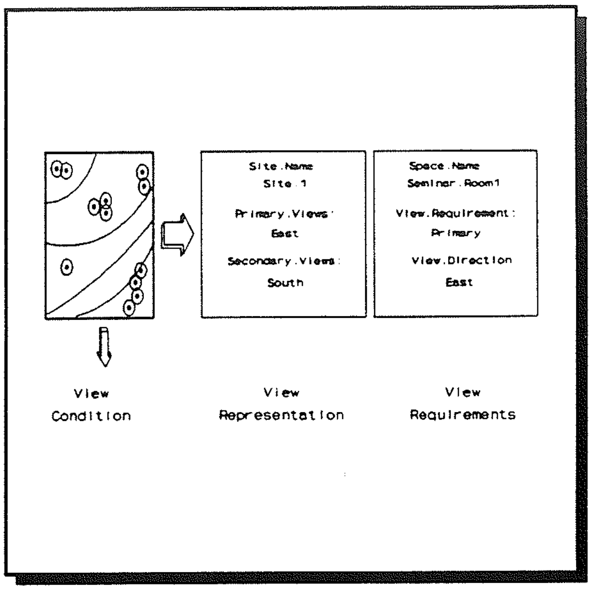


Figure 6-4: Translating actual view orientations to specific space requirements

Adjacency requirements contain a similar need to retain higher level guidelines due to the dependence on space definitions. The lack of prior knowledge concerning the exact set of spaces required in a layout prohibits the knowledge model from retaining adjacency guidelines focusing on specific spaces. Rather, the spaces inherit adjacency requirements for subclasses of spaces. For example, faculty offices may inherit the requirement for placement near other faculty offices. The analysis knowledge source transforms these class requirements to specific adjacencies by determining the existence of actual subclass instantiations. In the case of the faculty offices, if two faculty offices are defined, *faculty office 1* and *faculty office 2*, then the knowledge source places the appropriate adjacency requirement in each faculty office instance. Thus, the system retains the flexibility to alter the adjacency requirements according to the space definitions.

Constraint Removal

The final stage in generating space-specific requirements employs analysis rules to remove constraints deemed unnecessary based on design heuristics. For example, if the design program defines ten faculty offices, this does not indicate that each faculty office should be adjacent to the nine other faculty offices. Rather, each faculty office should be adjacent to an appropriate number of other faculty offices according to the particular problem instance. The CAADIE system performs this constraint removal activity by invoking heuristics to analyze and adjust the previously generated constraints. It should be noted, that these heuristics are the least developed within the CAADIE knowledge base. The CAADIE research effort has found this expertise to be the most dependent on individual designer preferences. Thus, to eliminate an apparent knowledge base bias, these heuristics are included strictly to illustrate constraint removal possibilities. However, upon completing this process, a complete set of typical requirements exists for each defined space.

Synthesis

The completion of the analysis phase prompts CAADIE to focus on synthesizing the general and space-specific requirements into space relationship diagrams. Synthesis occurs in either a system-based mode, or a user initiated mode. The former method instructs the system to generate configurations based on layout requirements. The latter method permits the designer to generate configurations with the assistance of CAADIE feedback.

System Initiated Layout Generation

The primary goal of the system initiated mode is to retain a balance between the general and space-specific requirements. This balance ensures that layouts contain both an underlying organization, and a recognition of the individual requirements impacting each space. Consequently, configurations will illustrate the trade-offs necessary to address the design program requirements.

CAADIE achieves this balance by incorporating a strategy which emphasizes hierarchical influence. In this strategy, the system obtains the greatest impact from designer preferences and design heuristics by utilizing these guidelines to reduce the number of placement locations examined for compliance with design program requirements. Rather than randomly generating placement options and then checking these options for design concept reinforcement and attribute compliance, the system sequentially generates placement options according to the preferences specified in the design heuristics. Subsequently, these options are analyzed for design attribute compliance. Thus, the first option generated which complies with the space-specific requirements can be retained with the understanding that the option is the most preferred in terms of reflecting both the underlying design concept and the space-specific requirements.

Figure 6-5 illustrates the hierarchical strategy by depicting the role each preference and requirement plays in the layout generation process. At the top level, the design concept determines the layout generation heuristics which will be used to generate placement options. This selection eliminates the possibility of the system generating options which do not reinforce the design concept. Second, the focus attribute provides a basis for selecting spaces to place in the configuration. The selection of spaces according to the focus attribute reinforces the philosophy of generating zones within the building. Third, the spatial ordering heuristics determine the preferred placement locations for each space. This determination ensures that the locations selected for each space will demonstrate established procedures for applying design concepts. Finally, the space-specific requirements validate the generated locations in accordance with the individual requirements of each space. This validation balances the design concept preferences with space-specific requirements by eliminating any generated options which do not comply with the stated requirements.

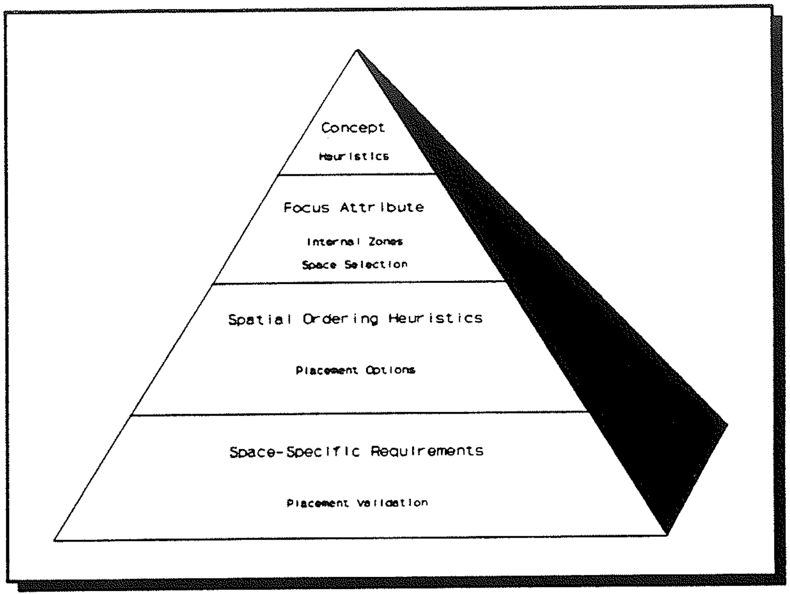


Figure 6-5: The hierarchical influence strategy incorporated in the system initiated mode

System initiated generation requires the cooperation and interaction of several elements including spatial ordering heuristics, placement refinement heuristics, placement evaluation functions, and the previously produced general and space-specific requirements. To illustrate this process, several decision trees and flow charts will be used as a basis for describing the interactions and process flows occurring in each layout generation phase. Figure 6-6 illustrates the primary steps and components within the synthesis process.

Space Selection

The space selection stage determines the order for spaces to be placed in the configuration. In contrast to mathematical-based systems which predominantly rely on static procedures such as the selection of spaces based completely on total area

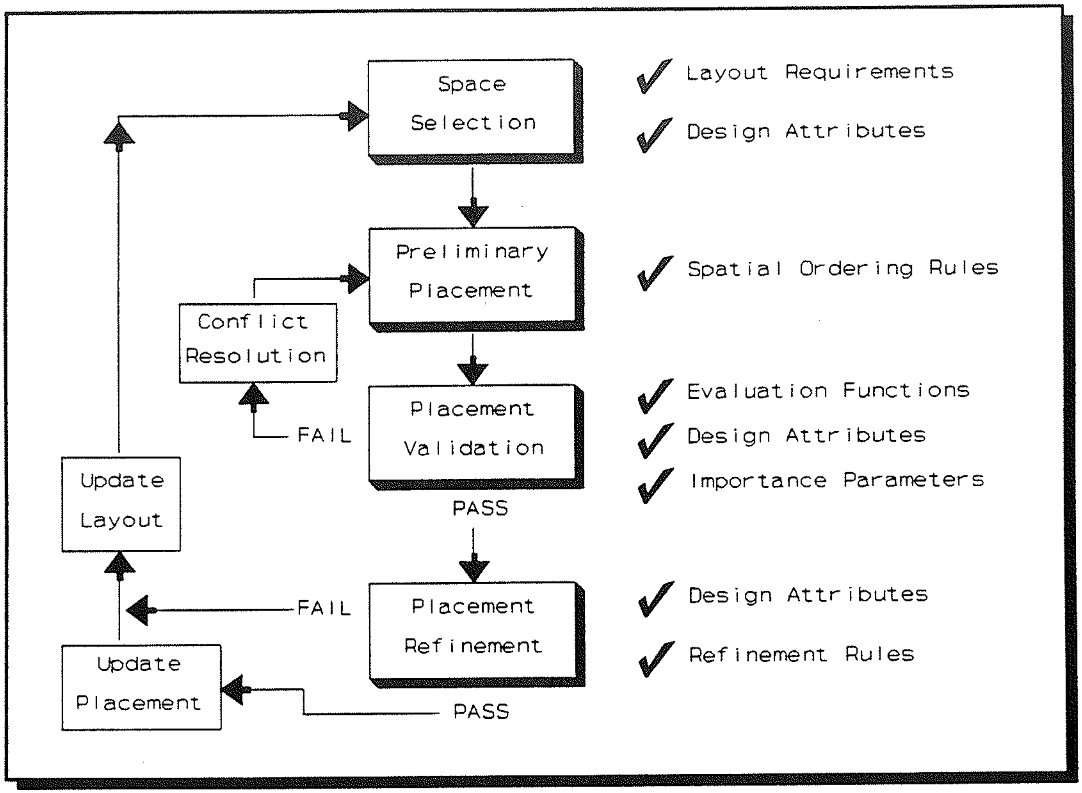


Figure 6-6: The system initiated synthesis process

requirements or adjacency requirements (Henrion 1978), CAADIE incorporates overall layout preferences and design attributes into a dynamic selection process. These attributes expand space selection into a dynamic process by combining the requirements of each successive space with the current layout conditions.

Space selection incorporates overall focus attribute and selection preference requirements, as well as, several space-specific requirements related to adjacencies and the focus attribute. Figure 6-7 illustrates this interaction with a decision tree representation. Within the decision tree, each node represents a yes or no question for which the system determines an answer based on the current configuration. The response to each question ascertains whether a more detailed question should be examined, or whether a selection decision can be achieved. Several key decision process characteristics which occur frequently throughout the decision nodes are as follows:

- The use of focus spaces - A key concept in the placement and selection of spaces is the use of focus spaces. A focus space is the space which the system has selected, prior to examining the remaining requirements, with which the system will attempt to place the next space adjacent. In the cases where the current focus space contains positive or desired adjacency requirements, these requirements guide the selection of spaces for placement (i.e., nodes 3, 7, and 11). In the cases where the current focus space does not contain remaining adjacency requirements, the selection process initially focuses upon selecting a new focus space (i.e., nodes 2, 7, 10, 11).
- Emphasis on interior zones - As previously discussed, the use of interior zones provides an underlying rationale for the configuration. To reinforce and facilitate zone development, the selection process emphasizes the selection of focus spaces and placement spaces containing similar attribute values as those featured in the current zone. For example, if privacy is the focus attribute, and the *private* zone is currently being emphasized, then spaces containing a requirement to be placed in the *private* zone will be emphasized over those containing other zone requirements (i.e., nodes 2, 4, 5, 6, 10, 11).
- Use of selection preference - Based on zone and adjacency requirements, several spaces may concurrently be candidates for selection as the next focus space or the next space for placement. When these situations arise, the final selection decision is based on the designer preference for selecting large or constrained spaces (i.e., nodes 2, 7, 8, 9, 10, 11, 12, 13). If large spaces are emphasized, then the space requiring the greatest square footage will be chosen. Similarly, if adjacency constrained spaces are emphasized, then the space containing the greatest number of adjacency constraints will be selected. Although these selection criteria are static attributes, they are only relied upon

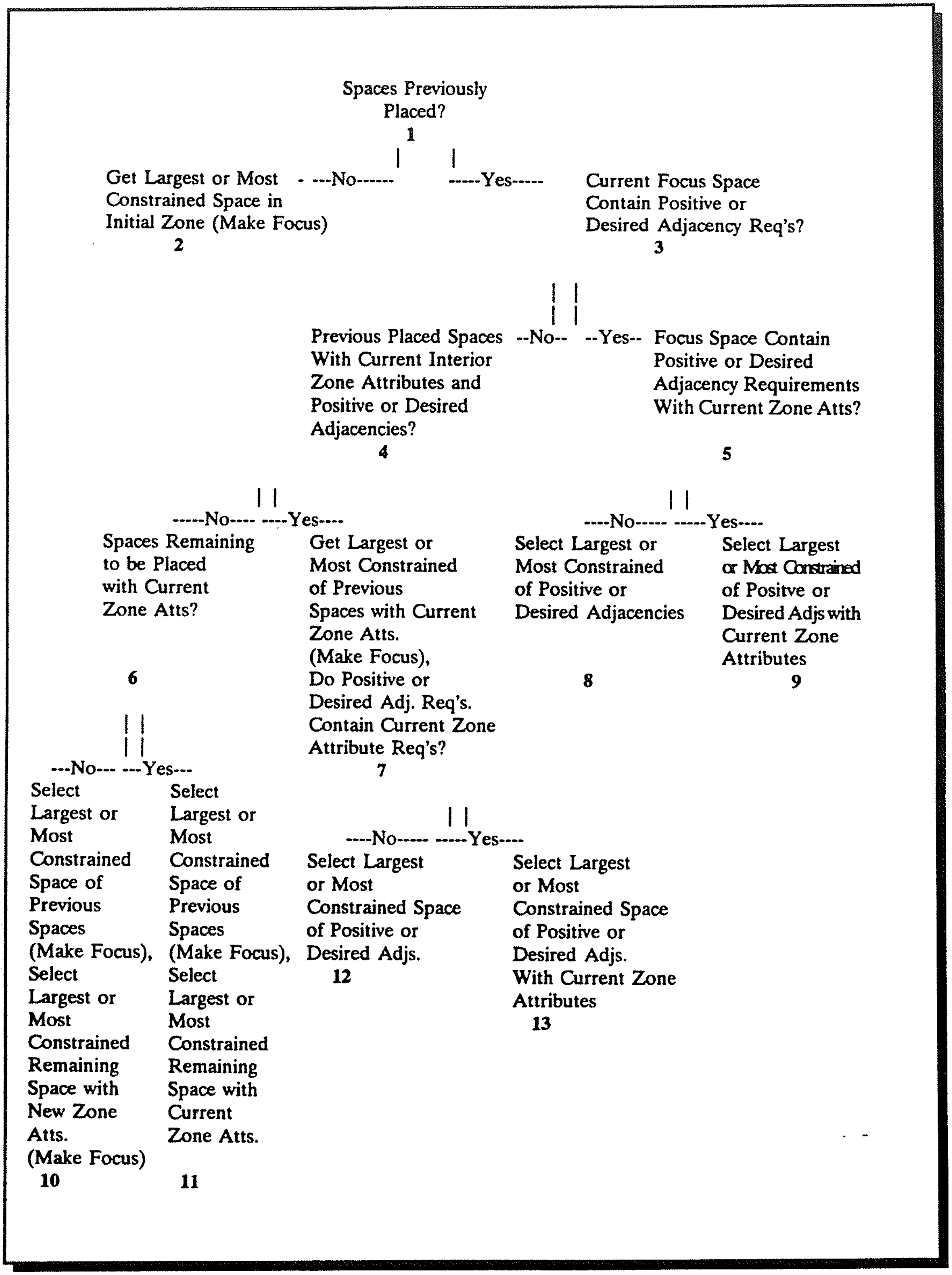


Figure 6-7: The decision tree used to select spaces for placement

when all preceding attributes have failed to yield a single space as the ideal selection.

At the conclusion of the space selection phase, three necessary components exist in preparation for preliminary placement: a focus space, a space for placement, and an interior zone to be developed.

Preliminary Placement Generation and Validation

Once a space is selected for placement, preliminary placement generation occurs through an iterative process combining the use of spatial ordering heuristics and placement evaluation functions. The spatial ordering heuristics characterize the preferences and fundamental design principles designers employ to reinforce specific design concepts. The placement evaluation functions validate placement options by detecting space-specific requirements and buildable area violations. The combination of these components ensures a balance between the design preferences represented in the heuristics and the constraints explicitly given in the layout requirements.

The initial step activates the spatial ordering heuristics associated with the currently selected design concept. Upon activation, the heuristics are analyzed to determine preferred placement options. This determination is made based on space-specific requirements and design concept situations captured in each placement heuristic. For example, the following linear concept rule balances main axis emphasis, with required spatial adjacencies.

Main Axis Preference Rule
IF The next space for placement has a required adjacency with the focus space
AND The adjacent sides of the focus space along the main axis are occupied
AND The adjacent side of the focus space on the other side of the main axis is unoccupied
THEN
Attempt to place the next space adjacent to the focus space on the other side of the main axis.

This rule portrays a preference for clustering spaces with required adjacencies along the main axis when the opportunity exists. Each of the remaining rules accordingly attempts to emphasize particular design principles to the greatest extent possible given the existing layout requirements.

Figure 6-8 depicts the generation of a placement option for a linear-based configuration. In this figure, *space 1* represents the focus space. Given a space-specific requirement for the next space to be located adjacent to the focus space, and the linear concept heuristics, the preferred placement option generated is illustrated by *space a*. The fulfillment of the heuristic conditions by this location, results in this location being considered as the preliminary placement option.

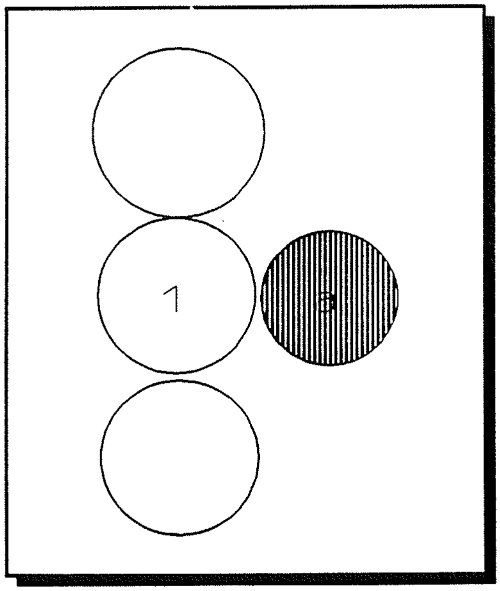


Figure 6-8: A preliminary placement option

The preliminary placement location represents the preferred position, however, this option requires validation to determine its compliance with the current space-specific requirements. If the designer indicates a global *important* or *desired* attribute preference, or an individual space located in the configuration contains an *important* or *desired* design attribute value, then relevant evaluation functions are invoked to determine requirement compliance. For example, if the focus space contains an *important* daylighting preference, then the daylighting knowledge source is invoked to detect the relevant requirement violations.

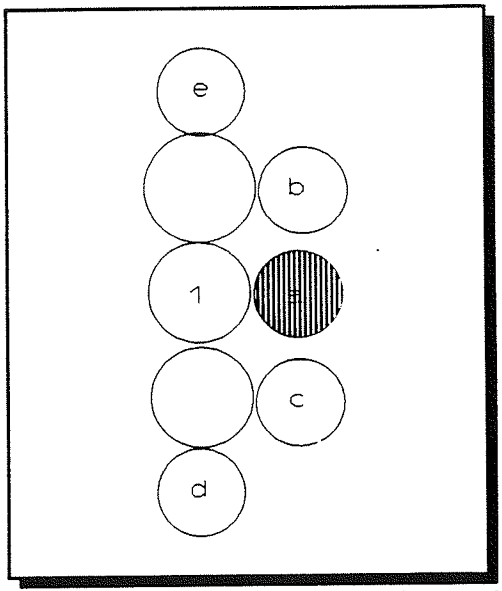


Figure 6-9: The generation of alternative placement options

This validation stage retains a balance between the implicit design preferences in the spatial ordering heuristics, with the explicit layout requirements. This balance results in layouts which demonstrate both sound design principles and a grounding in the reality of the design program requirements. In reference to the current example, if *space a* violates a space-specific requirement of *space 1*, or violates the buildable area restrictions, then the spatial ordering heuristics are reinvoked to generate new placement options. These new options will contain a reduced preference in terms of reinforcing the design concept, but will potentially comply with the stated space-specific and buildable area requirements. Figure 6-9 illustrates this iterative process, with *spaces b-e* representing the succession of generated placement options. The successful compliance of one of these positions with the layout requirements will conclude the preliminary placement phase, and result in the retainment of the placement option.

Conflict Resolution

While it is preferred that a placement option comply with all layout requirements, situations arise where the existing requirements create an over-constrained problem. Based on conflicting space-specific and buildable area requirements, it becomes impossible to comply with the complete range of design attributes. In these situations, it is necessary to relax certain requirements to resolve conflicts, and achieve a placement option which represents a compromise between the existing requirements.

Previous approaches to conflict resolution in computer-aided design systems have included the use of expert systems (Pohl et al. 1989), the separation of constraints into subgroups (McBrien, Madden, and Shadbolt 1989), and the use of designers as conflict resolvers (Akin, Dave, and Pithavadian 1988). Each of these approaches has been developed to emphasize either a delayed commitment or early commitment strategy to conflict resolution.

Delayed Commitment. - The delayed commitment strategy delays conflict resolution by retaining multiple placement alternatives. For example, in figure 6-9, rather than eliminating placement options until one is found which complies with all of the specified layout requirements, each placement alternative is retained as a partial configuration. Thus, the five partial configurations illustrated in figure 6-10 are each retained for further development. The advantage of this approach is that a decision as to which option should be selected can be delayed until later in the generation process. This delay permits the system to evaluate each of the options in the context of a further developed layout. Subsequently, a conflict which exists between two spaces in one of the configurations, may be overlooked due to the placement option being preferred in terms of the overall layout.

The disadvantage associated with this strategy is the potential proliferation of partial configurations (Mitchell, Steadman, and Liggett 1976). As each space is

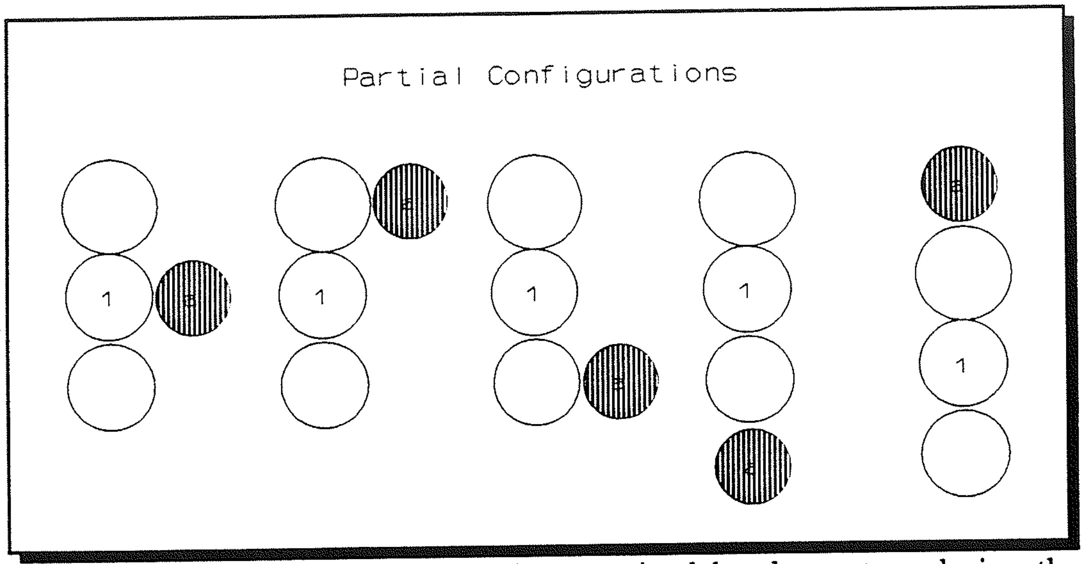


Figure 6-10: The partial configurations retained by the system during the delayed commitment strategy

added to the layout, and the resulting options are retained, the number of partial configurations continues to expand. Theoretically, the capabilities of computers should be almost unlimited in terms of addressing multiple alternatives and multiple issues within each alternative. However, the brute force approach required to analyze every alternative at every layout generation stage, makes it prohibitive to retain every alternative throughout the delayed commitment strategy (Levitt et al. 1989). In response to this issue, secondary mechanisms must be incorporated to periodically reduce the number of partial configurations currently retained by the system. Several approaches including both algorithmic (Flemming et al. 1988) and heuristic (Tommelein 1989), have been implemented to address this issue. However, the underlying concerns as to when the reduction should occur, and how the partial configurations should be evaluated, remain as open research issues.

Early Commitment. - The early commitment strategy contrasts with the delayed commitment strategy by focusing on the development of a single configuration throughout the layout generation process. Based on cognitive limitations, humans are forced to incorporate this strategy as a means to reduce the number of alternatives which must be examined during the layout generation process (Tommelein 1989). As placement options are generated for each space, only one option is retained at each step in the process. Thus, the layout problem is divided into a series of placement problems which can be addressed individually by the designer.

Computer-aided design systems emulate this process by relaxing constraints according to either design heuristics or predetermined algorithmic procedures, until, a single option can be generated which complies with each of the remaining constraints. This option is then retained by the system as the selected placement location. In addition to eliminating the multiple configuration issue, the early commitment strategy retains the advantage of ensuring that a partial solution has been achieved at the conclusion of each step in the process. The relaxation of constraints permits the system to explore placement options until, in the extreme case, no constraints remain which impact the current space. Thus, the system pursues placement options until a solution is found. Consequently, the system eliminates the requirement to reevaluate the partial configurations at the conclusion of the layout generation process.

The disadvantage associated with the early commitment strategy is the loss of the opportunity to view placement options in the context of a completed layout. Each placement option is limited to a local evaluation related to the current configuration state. Subsequently, an option which appears to be the best option at the time it is generated, cannot be reevaluated later in the generation process. This may have a significant impact on the layout in situations where the selection of a less preferred local option may have resulted in producing a better overall configuration.

The CAADIE approach to conflict resolution is built on an early commitment strategy. The determination to utilize this strategy is founded on the CAADIE approach of using design principles and heuristics to generate placement options. In accordance with this approach, the conflict resolver incorporates heuristics which guide the conflict resolution process. Based on these heuristics, the system ensures that the designer preferences and space-specific requirements are addressed by each selected placement option. Although this approach retains the problem of local versus overall design context, the assumption is made, that significant knowledge is contained within the system to generate preferred locations for each space which will combine to form a sound configuration.

Figure 6-11 illustrates the decision tree used to guide the conflict resolution process. Based on the current design state, the questions are used to determine which attributes should be relaxed through the duration of the placement process associated with the current space. The following factors influence the conflict resolution process:

- A bias toward external attributes - An emphasis on any specific area of constraints creates a bias towards complying with certain requirements prior to complying with others. This bias occurs in the CAADIE conflict resolver when the determination to comply with attributes cannot be based entirely on attribute preference factors. In these instances, the conflict resolver contains a bias towards complying with external constraints such as views and daylighting. This is due to external resources being in limited supply, and failure to comply with these constraints requires a greater compensation cost than failure to comply with internal constraints. For example, daylighting compensation requires either additional artificial lighting or secondary daylight sources. This could be a notable increase over the compensation required for lack of compliance with an internal constraint such as privacy.

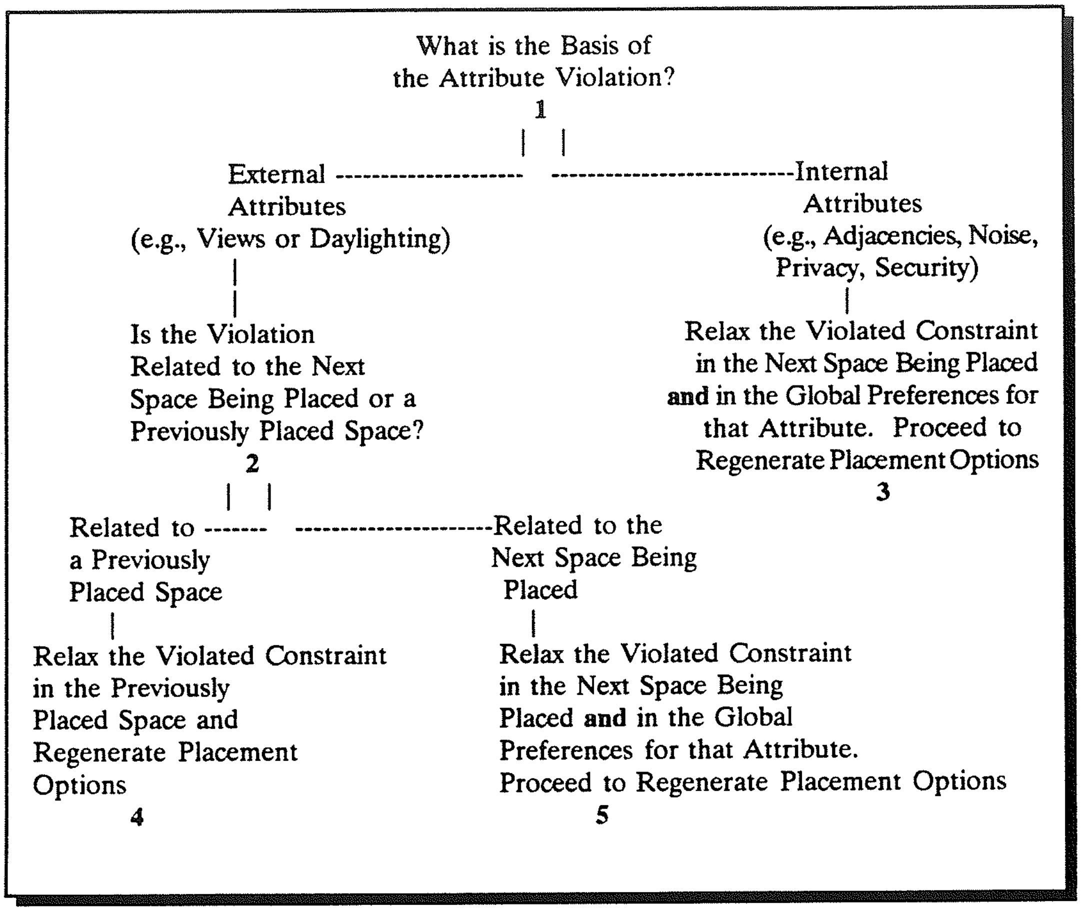


Figure 6-11: The decision tree utilized in the conflict resolution process

- Previously placed spaces have priority - The CAADIE system generates layouts through a continuous aggregation of intelligent placement decisions. In this strategy, it becomes judicious to preserve previously satisfied requirements such as view and lighting requirements, to ensure continued design program compliance. Thus, when given the option to relax a requirement associated with a previously placed space, or one associated with the space being placed, the conflict resolver will relax the constraint associated with the space being placed.
- Relax desired constraints of the space being placed, prior to relaxing important constraints - The attribute importance scheme incorporated in the

CAADIE system guides the conflict resolver in analyzing the space being placed for attribute compliance. Based on the attribute preferences, compliance with *desired* attributes is determined prior to determining compliance with *important* attributes. By analyzing *desired* attributes prior to *important* attributes, the system will initially detect violations of *desired* attributes. This detection permits the system to record these attributes as the first ones which should be relaxed. This strategy ensures the opportunity to relax attribute requirements in accordance with the existing attribute importance factors.

The result of this conflict resolution process is a new set of conditions for the generation of placement options. These new options are evaluated in accordance with the new attribute preferences, resulting in less stringent requirements for attribute compliance.

Placement Refinement

The final synthesis step refines the preliminary placement option according to internal design relationships. This process capitalizes upon the flexibility remaining in the preliminary placement option to create further adjacency relationships. As illustrated in figure 6-12, the requirement to place *space 4* adjacent to *space 1* provides the flexibility to place *space 4* either directly on the east side of *space 1*, or anywhere along the circumference of *space 1* between *space 3* and

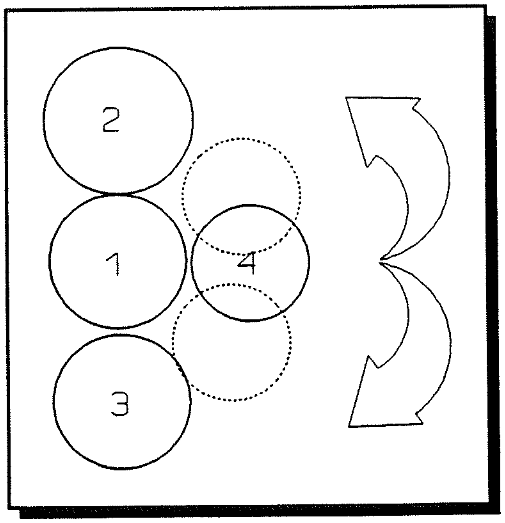


Figure 6-12: Placement refinement options

space 2. The refinement process determines the potential to alter the initial placement such as that illustrated by *space 4*, in an effort to take advantage of relationships with spaces located adjacent to the focus space, such as *spaces 2 and 3*.

The existence of design relationships is determined based on the internal design attributes given for each space. The design relationship definition is given as follows:

> If two spaces have similar requirements for an internal design attribute such as security or privacy, and this requirement is *important* to at least one of the spaces and at least *desired* to the other, then these two spaces have a design relationship.

If a design relationship exists, then CAADIE attempts to support this relationship by adjoining the two spaces. However, the new adjacency cannot violate any layout requirements, or break any existing adjacencies (figure 6-13). The detection of a design relationship between *spaces 4 and 2* prompts CAADIE to analyze the potential of adjoining *space 4* with both *spaces 1 and 2*. If this location complies with all layout requirements, then the new location is retained in the configuration. At this point, the synthesis cycle returns to the selection stage, and repeats until all spaces are placed in the configuration.

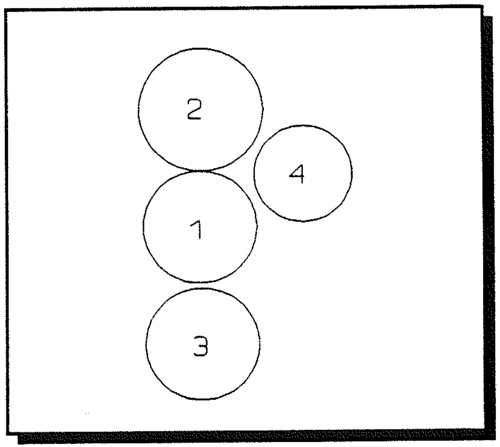


Figure 6-13: The conclusion of the placement refinement process

User Initiated Layout Generation

The second synthesis mode transfers the layout generation focus from the knowledge sources to the designer. In this mode, the designer manually develops configurations with complete placement flexibility. Based on this flexibility, the system transforms its function from a layout generator to an electronic sketch pad. This analogy accentuates the system's role as a tool to either explore ideas and concepts, or obtain further problem clarification through evaluation feedback.

The sketch pad mode diverges from the system initiated mode by operating in an iterative, three-step process comprising space selection, space placement, and evaluation feedback generation. These steps facilitate designer initiated generation within a participatory design environment.

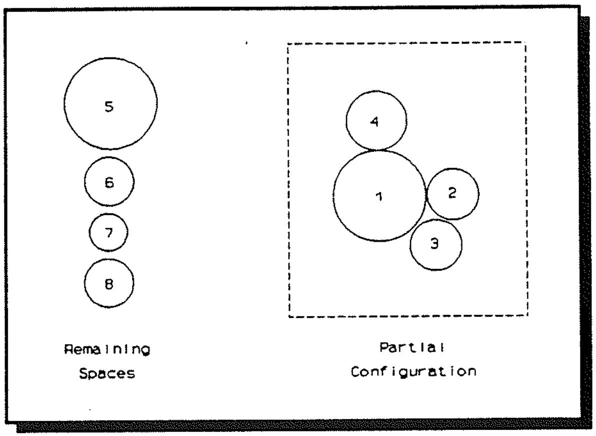


Figure 6-14: Space selection options in the user initiated mode

Select a Space

The space selection step requires the designer to select a space for placement from either the list of remaining spaces, or the current partial configuration (figure 6-14). The selection of a remaining space indicates an intention to add a new space to the configuration. The selection of a previously placed space indicates a preference for altering the existing configuration. The system provides decision support for either selection by displaying the respective space-specific requirements.

Place the space

After selecting a space, the designer has the option to either place the space adjacent to, or overlapping, the configuration (figures 6-15 and 6-16). A spatial overlap prompts CAADIE to resolve the situation by altering the location of affected spaces. This movement occurs through an algorithmic approach stressing the alteration of space locations by the minimum distance possible. The assumption behind this algorithm is that the less a space is moved, the greater the opportunity will be to retain the existing requirement compliance. Figure 6-17 illustrates the result of resolving the overlap condition initiated by the previous placement.

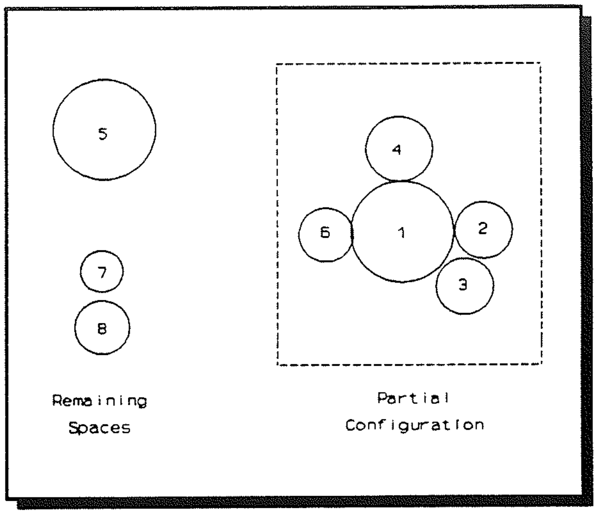


Figure 6-15: The manual placement of a space to adjoin the partial configuration

At the conclusion of the movement stage, a condition could exist where a space remains "floating" (i.e., located in a position not adjoining any other space). In these cases, CAADIE attempts to relocate the floating space according to the space-specific adjacency requirements. If the location dictated by these requirements does not violate any of the remaining design requirements, then the space

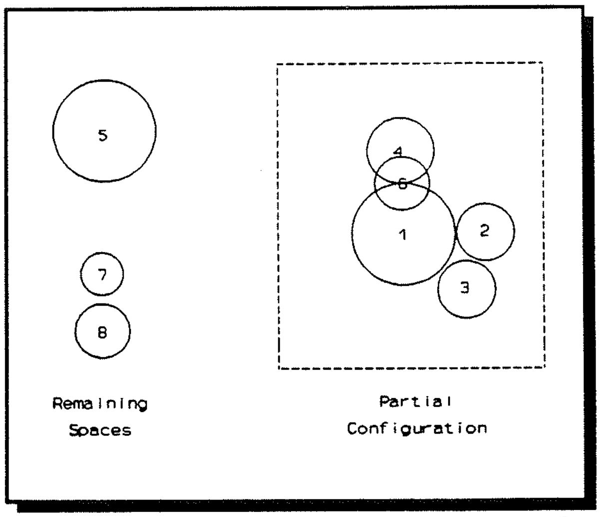


Figure 6-16: The manual placement of a space which overlaps the current configuration.

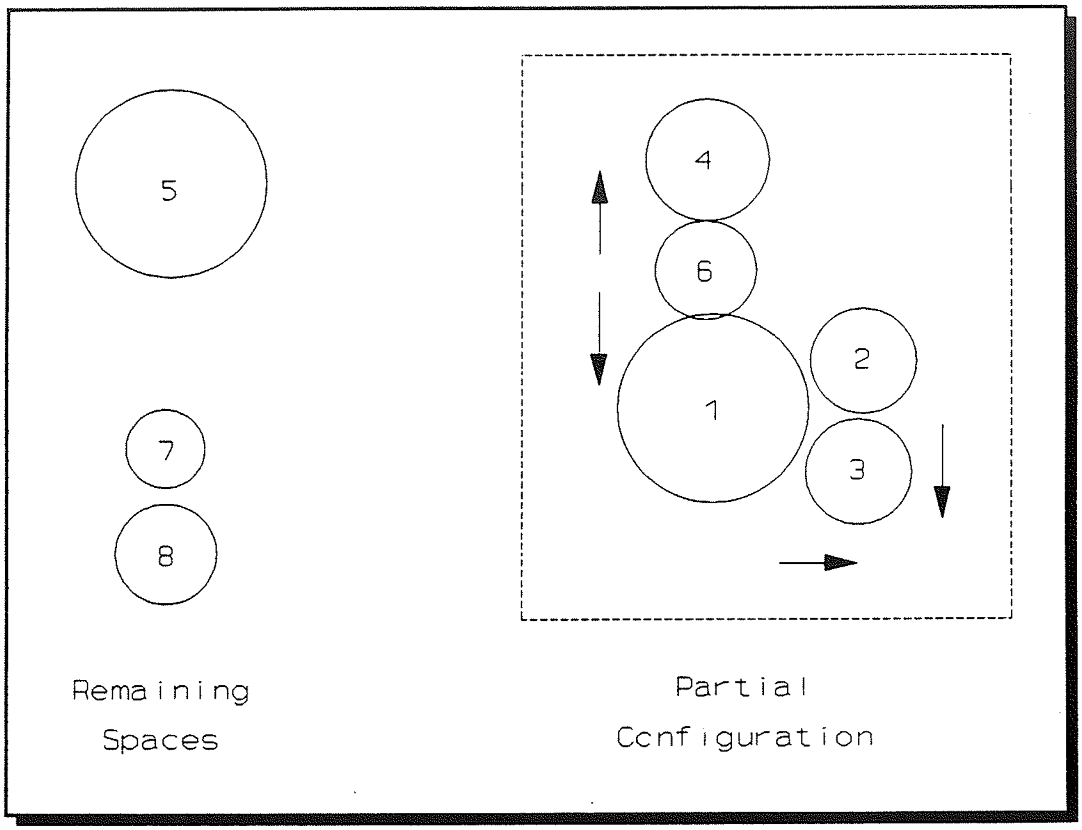


Figure 6-17: Resolution of the spatial overlap condition

regains relationship with the configuration. If this requirement compliance cannot be met, then responsibility is returned to the designer for adjoining the space.

Evaluation Feedback

The final placement step provides the designer with feedback based on requirement compliance (this stage is described in detail in the following section). Given this feedback, the designer may choose to alter the placement of a space in an effort to comply with specific requirements. The iteration between this movement-feedback process provides the opportunity to reevaluate and refine design decisions.

Evaluation

The final layout generation stage encompasses the overall evaluation of configurations. This evaluation process provides a quantitative measurement based on requirement compliance. This objective measurement removes the biases associated with subjective evaluations based on issues such as aesthetic quality, which are susceptible to individual designer interpretations. By removing these interpretations, the measurement provides a baseline to compare the overall compliance ratings of layout alternatives. These comparisons may then be combined with the designer's own subjective interpretations and experience to ascertain appropriate layout alternatives.

Historically, this stage contains the dominant problem of determining how the evaluation of configurations should be accomplished, and what factors should be included in the evaluations. Initially, the measurement focused on optimization factors such as spatial distances (Mitchell 1977). In this form of evaluation, configurations containing the minimum total distance between spaces requiring materials and people to be moved between them were considered the best solutions. Although this technique may be viable for circulation constrained problems such as industrial plants and warehouses, the assumption that circulation will always be the overriding design issue reduces the applicability of this technique.

Evaluation procedures have since shifted to emphasize the inclusion of multiple design attributes in the overall evaluation measurement (Schmitt 1988b). These measurement procedures combine individual attribute compliance ratings into an overall layout rating. The CAADIE evaluation scheme follows this multi-attribute trend by evaluating configurations based on the complete range of requirements given for each design program. Additionally, this evaluation process reflects designer preferences by including attribute importance factors into the evaluation ratings. Figure 6-18 illustrates the evaluation process included in the CAADIE system.

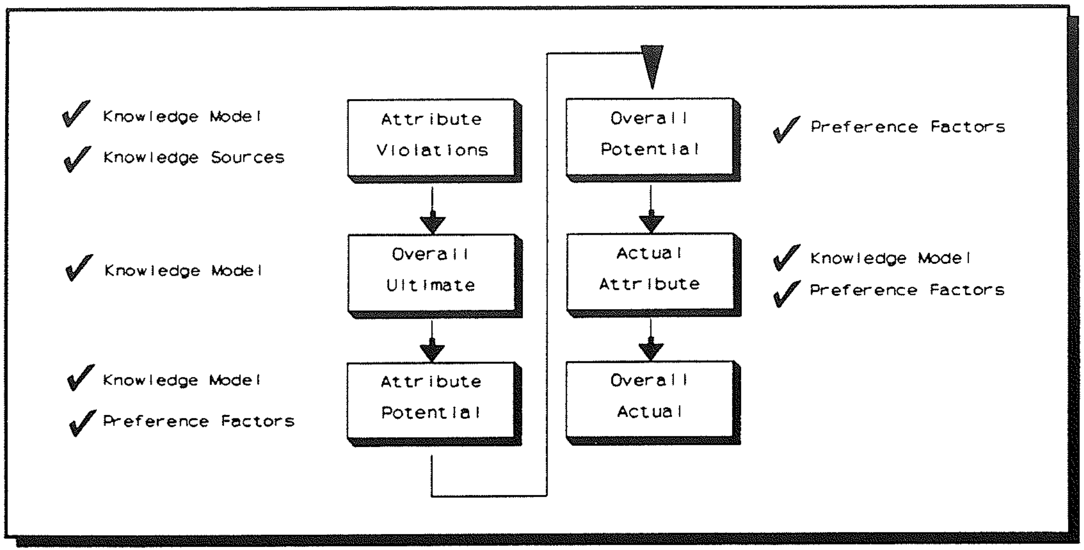


Figure 6-18: The layout evaluation process including evaluation stages and influences

Unrefined Attribute Violations

The initial evaluation step entails analyzing a configuration to detect space-specific requirement violations. This analysis is executed by knowledge sources associated with respective design attributes. Each knowledge source contains the appropriate functions to detect violations in accordance with the unique requirements of each attribute. For example, the daylighting knowledge source detects lack of external access to spaces with daylighting requirements, while the acoustics knowledge source detects conflicts between spaces with high noise output and spaces with low noise tolerance. Based on these evaluation functions, the total number of violations are accumulated for each design attribute. These numbers represent an unrefined evaluation based entirely on the number of configuration violations. This initial evaluation is intended to provide designers with a secondary level of feedback from which to make design decisions.

Overall Ultimate Score

Once the requirement violations are detected, the evaluation procedure emphasizes the generation of a weighted evaluation incorporating both the violations and the attribute importance factors. This process requires calculating the overall number of points ultimately achievable for a given configuration. This number represents the total number of points achievable if every attribute contained an *important* attribute importance factor. Subsequently, this total provides the basis for generating a scaled 0-100% evaluation, where 100% indicates every attribute contains an *important* attribute importance factor. Any circumstance where less than every attribute is *important*, will result in a less than 100% rating.

At this point, it is necessary to digress slightly into the underlying motivation behind the 0-100% scale, and the method for integrating attribute importance factors into the point calculations. The weighted scale evolves from designer tendencies to alter configuration evaluations according to the importance of different attributes. Specifically, the failure to comply with attributes containing lower importance factors is not penalized as severely as violations of attributes containing higher importance factors. Concurrently, compliance with lower rated attributes does not count as strongly in the overall rating as compliance with higher rated attributes. Thus, in calculating the overall layout rating, the existence of attributes with *neutral* or *desired* importance factors reduces the overall potential layout rating. This reduced rating is reflected by a reduction on the 0-100% scale.

The integration of these importance factors is accomplished through an underlying point system. If an attribute is important to a given space, then the number of possible points for that attribute is calculated by multiplying three times the number of attribute compliance possibilities for that space. The number of these compliance possibilities will vary for each attribute. For example, a view requirement is either achieved or violated based on external access to the appropriate space.

Thus, for each space, one compliance possibility exists for a view requirement. In contrast, security requirements for a specific space may be violated by any adjacent space. This results in the number of compliance possibilities being equal to the number of adjacent spaces. When multiplied with an *important* importance factor, the preceding view requirement has a potential of receiving three points, while the security requirement has a potential of receiving three times the number of adjacent spaces. Similarly, desired attributes are calculated by multiplying two times the number of compliance possibilities, and neutral attributes receive only the number of compliance possibilities (equation 6-1).

This point allocation scheme is not a final solution to the evaluation problem. However, until extensive studies are conducted on the methods designers use to differentiate between various layout influences, a substitute method is required to provide designers with layout ratings. In the absence of these studies, a point allocation method represents one opportunity for presenting overall layout evaluations. Therefore, the CAADIE scheme has been incorporated for its potential to provide a numeric basis from which designers may compare various configurations. More importantly, it provides a consistent evaluation basis for every layout generated within the system. Given the consistency of this evaluation scheme, the actual multiplication factors used to generate the rating are less significant. The necessary element being that, the designer is presented with an evaluation rating reflecting the importance factors of each attribute, and the designer understands the evaluation rating implications.

Given this background, the procedure for calculating the overall ultimate score is summarized by the following equation:

$$\text{Overall Ultimate Score} = \sum_{i=1,\, j=1}^{n} (\text{Attribute Compliance Possibilities}_{(i,j)} * \text{Important Preference Factor}_{(i,j)})$$

$$\text{Where } i = \text{spaces},\ j = \text{attributes},\ n = \text{number of spaces}$$

Equation 6-1: Overall Ultimate Score

Potential Attribute Points and Percentages

The number of potential attribute points represents the total number of points an individual attribute could obtain with complete requirement compliance. This number is weighted by incorporating either the attribute preferences existing in each space, or the global attribute preferences, through the point allocation method previously discussed. The relevant equation used for each attribute is as follows:

$$\sum_{i=1}^{n} (\text{Attribute Compliance Possibilities}_i * \text{Importance Factor}_i)$$

$$\text{Where } i = \text{spaces},\ n = \text{number of spaces}$$

Equation 6-2: Potential Attribute Points

The potential attribute percentage follows from the potential point total by indicating the relationship of each attribute to the sum of the attribute point totals. This scaled number provides feedback highlighting the potential impact of altering the layout to comply with individual attribute requirements. The designer may then determine the trade-offs between altering attributes containing a greater or less impact on the overall rating. The equation for each attribute based on the aggregation of all potential attribute points is as follows:

$$\text{Potential Attribute } \%_i = \frac{\text{Potential Attribute Points}_i}{\sum_{i=1}^{n} \text{Potential Attribute Points}_i}$$
$$\text{Where } i = \text{attributes}, \ n = \text{number of attributes}$$

Equation 6-3: Potential Attribute Percentage

Overall Layout Points and Percentage

After obtaining the potential number of points for each attribute, the evaluation process returns to calculating the overall layout potential. This calculation incorporates the attribute importance factors for each attribute. The inclusion of these factors is accomplished by building upon the previously completed individual attribute calculations.

$$\text{Overall Layout Points} = \sum_{i=1}^{n} \text{Potential Attribute Points}_i$$
$$\text{Where } i = \text{attributes}, \ n = \text{number of attributes}$$

Equation 6-4: Potential Number of Layout Points

The result from this calculation provides the basis for calculating the scaled layout potential. Based on the number of potential points and the ultimate number of potential points, the scaled layout potential exists as the maximum rating the layout can achieve with 100% attribute compliance.

$$\text{Overall Potential \%} = \frac{\text{Overall Potential Points}}{\text{Overall Ultimate Points}}$$

Equation 6-5: Potential Layout Percentage

Actual Attribute Points and Percentage

The calculation for the actual number of attribute points diverges from the previous calculations by incorporating the detected attribute violations. These violations are included by utilizing a penalty factor corresponding to the attribute importance factors. The following equation is used for each attribute:

$$\begin{gathered}\text{Actual Attribute Points} = \\ \sum_{i=1}^{n} (\text{Potential Attribute Points}_i - \\ (\text{Attribute Violations}_i * \text{Attribute Importance Factor}_i)) \\ \text{Where } i = \text{spaces}\end{gathered}$$

Equation 6-6: Actual Attribute Points

Given the actual number of attribute points, the scaled attribute rating follows as a percentage of the potential attribute rating. The two-step process serves to: (1) obtain a relation between the attribute requirements met and the total number of attribute requirements, and (2) scale the percentage in accordance with the potential attribute percentage.

$$\textit{Attribute Compliance }\%_i = \frac{\textit{Actual Attribute Points}_i}{\textit{Potential Attribute Points}_i}$$
$$\textit{Where } i = \textit{attributes}$$

Equation 6-7: Actual Attribute Compliance Rating

$$\textit{Scaled Compliance }\%_i = \textit{Potential Attribute }\%_i * \textit{Attribute Compliance }\%_i$$
$$\textit{Where } i = \textit{attributes}$$

Equation 6-8: Scaled Attribute Compliance Rating

Overall Actual Percentage

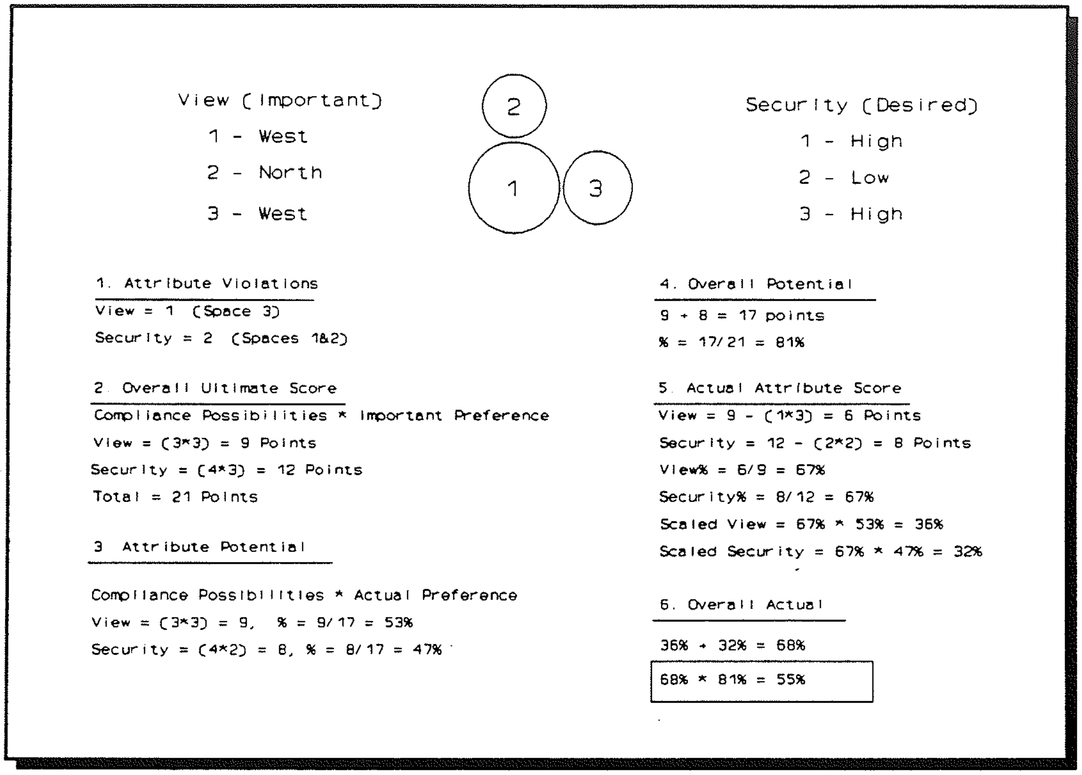
The final layout evaluation step entails generating the overall layout rating by multiplying the overall potential points with the sum of the individual attribute percentages.

$$\textit{Overall Layout Rating} = \left(\sum_{i=1}^{n} \textit{Scaled Compliance }\%_i\right) * \textit{Overall Potential Poin}$$
$$\textit{Where } i = \textit{attributes}$$

Equation 6-9: The Overall Layout Rating

Example

Figure 6-19 presents an example of the evaluation process for a simple, three space layout with two design attributes. For illustrative purposes, the attributes contain only global importance factors, *desired* for security and *important* for view. Based on these preferences, and the limited layout requirements, the example presents the six principal calculations performed in the evaluation process. To



View (Important)
1 - West
2 - North
3 - West

Security (Desired)
1 - High
2 - Low
3 - High

1. Attribute Violations
View = 1 (Space 3)
Security = 2 (Spaces 1&2)

2. Overall Ultimate Score
Compliance Possibilities * Important Preference
View = (3*3) = 9 Points
Security = (4*3) = 12 Points
Total = 21 Points

3 Attribute Potential
Compliance Possibilities * Actual Preference
View = (3*3) = 9, % = 9/17 = 53%
Security = (4*2) = 8, % = 8/17 = 47%

4. Overall Potential
9 + 8 = 17 points
% = 17/21 = 81%

5. Actual Attribute Score
View = 9 - (1*3) = 6 Points
Security = 12 - (2*2) = 8 Points
View% = 6/9 = 67%
Security% = 8/12 = 67%
Scaled View = 67% * 53% = 36%
Scaled Security = 67% * 47% = 32%

6. Overall Actual
36% + 32% = 68%
68% * 81% = 55%

Figure 6-19: An example of the layout evaluation process

reiterate, the basis for deriving the numeric value in this example remains consistent for each layout generated in the system and represents baseline feedback from which the designer can pursue and examine layout ideas.

Summary

The preceding sections portray the fundamental layout generation processes within the CAADIE prototype. These processes combine knowledge-based and algorithmic reasoning paradigms to produce or support layout generation. Several key points evident in the process form the central layout generation issues.

- The layout generation process is an iterative process, requiring the cooperation of analysis, synthesis, and evaluation components.

- The automated generation of space-specific requirements is essential in reducing designer input requirements, and ultimately, achieving designer acceptance.
- Capturing designer preferences for general design concerns is required to implicitly capture designer experience and intuition.
- Incorporating layout requirements and design concepts in the synthesis phase balances design principles with design program compliance requirements.
- Generating evaluation feedback based on consistent parameters provides a designer with baseline information from which to compare layout alternatives.

CHAPTER 7

USER INTERFACE

The common theme underlying the CAADIE project emphasizes the use of a participatory design environment. This emphasis evolves from an acknowledgement of the important role a designer plays in the layout generation process. In contrast to the majority of layout generation research, the CAADIE research views the designer as an integral system component. The experience and intuition a designer brings to a layout problem represents an influential force behind design evolution. To effectively capture this experience, the system should facilitate active designer participation by creating an environment where designers are able to impart knowledge, develop ideas, and dictate the degree of design assistance.

The emphasis on an interactive environment results in the designer retaining the prerogative to determine the system automation level. This level can range from complete automation, where the designer provides a minimum of general layout requirements, to a manual system, where the designer directly generates layouts. The primary component required to support this participatory design environment is the user interface.

User Interface Issues

Human-computer interaction plays a pivotal role in generating design solutions. Throughout the design process, from design program development to final solution refinement, designers impart and receive information through a user interface. The ability to effectively interact with this interface through input options and output facilities, will have a subsequent impact on the manner in which designers use the system. This underlying importance has prompted design researchers to address user interface issues as an integral research component. The objective being that, a successful user interface facilitates the exploration of alternative design solutions.

Central to the CAADIE development effort has been the identification of target users, and the design of appropriate graphic and knowledge representations. The following sections expound upon these issues and their role in user interface development. Following this discussion, the chapter introduces the CAADIE user interface elements and the relationship of this interface to the iterative layout generation process.

Who is the User?

The identification of target users is a fundamental step in developing an appropriate user interface. The requirements of architectural designers vary according to many factors including overall design experience, familiarity with specific building types, expertise in addressing particular design attributes, and familiarity with CAD systems. Appropriately addressing these factors within the user interface will significantly impact eventual system usage. Several studies reinforce this idea by identifying relationships between designer acceptance and the user interface emphasis on designer requirements (Eastman 1973; Akin 1978; Hyde 1989). Specifically, the studies show that user interfaces lacking a coherent designer focus, fail to retain a consistent level of difficulty. At the extreme, the interface developer will be the only user who can realistically interpret the system's feedback. As a result, the system will be incapable of reaching its true potential as a design assistant due to lack of user understanding, and ultimately, lack of user acceptance.

Given this relationship between user interface and designer acceptance, the CAADIE development effort addressed the target user issue as an initial project concern. During the knowledge acquisition process, discussions with design consultants defined a baseline user model and associated interface requirements. The following is a list of user requirements based on these discussions:

- Level of Expertise - The CAADIE user interface assumes the designer possesses at least a familiarity with principal layout generation issues. This familiarity implies that the designer can identify and select design concepts, comprehend the influence of design attributes on a layout solution, and acknowledge the necessity to relax attribute requirements during conflict resolution. The interface does not address the extensive explanation requirements of novice designers, nor the detailed design histories required by design experts. Rather, the interface supports the designer who addresses particular building types on a regular basis, and who requires assistance in the examination of layout issues and alternatives.
- Design Attribute Expertise - The CAADIE user interface targets designers who have experience in addressing design attribute requirements during the conceptual design of architectural floorplans. Based on this characterization, the user interface permits designers to examine the consequences of altering design attributes through evaluation summaries such as violation reports and graphic requirement representations. This level of feedback contrasts with the requirements of experts in areas such as lighting or acoustics, who require detailed feedback reports including relevant calculations, dimensions, and assumptions, to determine the appropriateness of design decisions. Thus, the CAADIE interface provides the opportunity to explore the influence of individual attributes on the layout generation process, however, detailed attribute analysis is deferred to specialized analysis programs.
- CAD System Familiarity - The CAADIE system is a prototype computer-aided design system. As a result, the system does not include the breadth of drawing and information access capabilities available in commercial CAD

systems. The lack of these capabilities does not hinder designers from utilizing fundamental options such as selecting spaces and indicating layout preferences. However, the absence of these options requires familiarity with the KEE environment to take advantage of advanced system features such as knowledge base modification. This limitation could be overcome in future implementations focusing on user interface concerns. However, these advanced user-oriented features are technical issues rather than research issues, and are outside the project scope and contribution.

Interface Style

The interface style impacts a designer's process by either diverging from, or reinforcing, the natural idioms and tools used during specific design process phases. The reinforcement of these idioms and tools permits layout development to occur in a familiar manner and context. To illustrate, during conceptual design, designers sketch numerous layout diagrams to explore layout alternatives. These diagrams and sketches are conceptual design idioms and tools respectively. Capturing these idioms and tools permits the continuation of established layout generation processes. In contrast, diverging from these idioms and tools forces a designer to incorporate new layout generation processes. The inherent risk in this approach is the possibility a designer will reject the system due to its incompatibility with normal practice.

The CAADIE response to these interface style issues has been to identify and include conceptual design idioms throughout the system interface components. Specifically, the user interface incorporates bubble diagram representations, histogram feedback charts, and graphic requirement representations, to reinforce typical layout processes. These idioms provide the necessary components to explore as many layout options as the designer deems necessary to reach a layout solution. Concurrently, the user interface is patterned after the common tools used during layout generation. As discussed previously, the interface has the potential to serve as an electronic sketch pad. Similar to generating sketches on multiple sketch pad pages, the use of a mouse and the interface tools, enables designers to interactively generate ideas and examine layout issues over the course of multiple layout alternatives. The combination of this sketch pad capability and the conceptual design idioms, serves to reinforce familiar layout generation processes.

Knowledge Interface

The knowledge interface issue emphasizes the influence of layout information on individual design objects. In contrast to computer-aided drafting systems, knowledge-based design systems contain information and knowledge beyond point-line descriptions. This extended information contains typical geometric and non-geometric requirements and guidelines for each design object. This information impacts the role each design object plays in the layout generation process by restricting the potential range of positions available for the object within a layout alternative. The impact of this information raises the issue of how the user interface supports access to, and presentation of, design object requirements.

Whereas the point-line representations of drafting systems provide minimal information beyond that provided by line drawings, the underlying information in a knowledge-based design system provides valuable design program perspectives. Access to this design program is necessary to provide capabilities such as exploring the ramifications of altering design object attributes, and manipulating layout attributes to pursue design alternatives. Previous approaches to this knowledge access issue include using a design notebook analogy (Bond et al. 1988), separate information displays showing the influence of specific attributes on design decisions (Pohl et al. 1991; Fenves et al. 1989), constraint displays illustrating attribute interrelationships (Flemming et al. 1988; Tommelein 1989), and direct design program access (Schmitt 1988b). In each of these approaches, the common objective is to provide an understanding of, and access to, the underlying requirements driving the generation of layout configurations.

The CAADIE project approach to this access issue accentuates an "intelligent object" access strategy. Within this approach, the designer accesses design requirements directly from the design objects influenced by the requirements. This strategy builds upon the overall system goal of placing layout requirements in the context in which they are used. If a designer wants to obtain information related to an individual design object, then the information is readily available from that object. The user interface facilitates this strategy by permitting all requirements to be obtained through the interactive selection of design objects. Upon selection, the user receives a listing of the design program requirements associated with that object. As discussed below, this strategy is enhanced by the ability to obtain graphic representations which illustrate the impact of individual requirements on the overall layout.

Interface Introduction

Figure 7-1 illustrates the user interface as it appears prior to the generation of space relationship diagrams. The principal interface components include: the diagram output window, the evaluation charts, and the general requirement windows. These components are divided into separate interface groups addressing particular design process stages. Specifically, the requirement windows facilitate analysis interaction, the output window portrays synthesis results, and the evaluation charts provide evaluation feedback. This separation is intended to provide an additional design context in which to place layout generation alternatives.

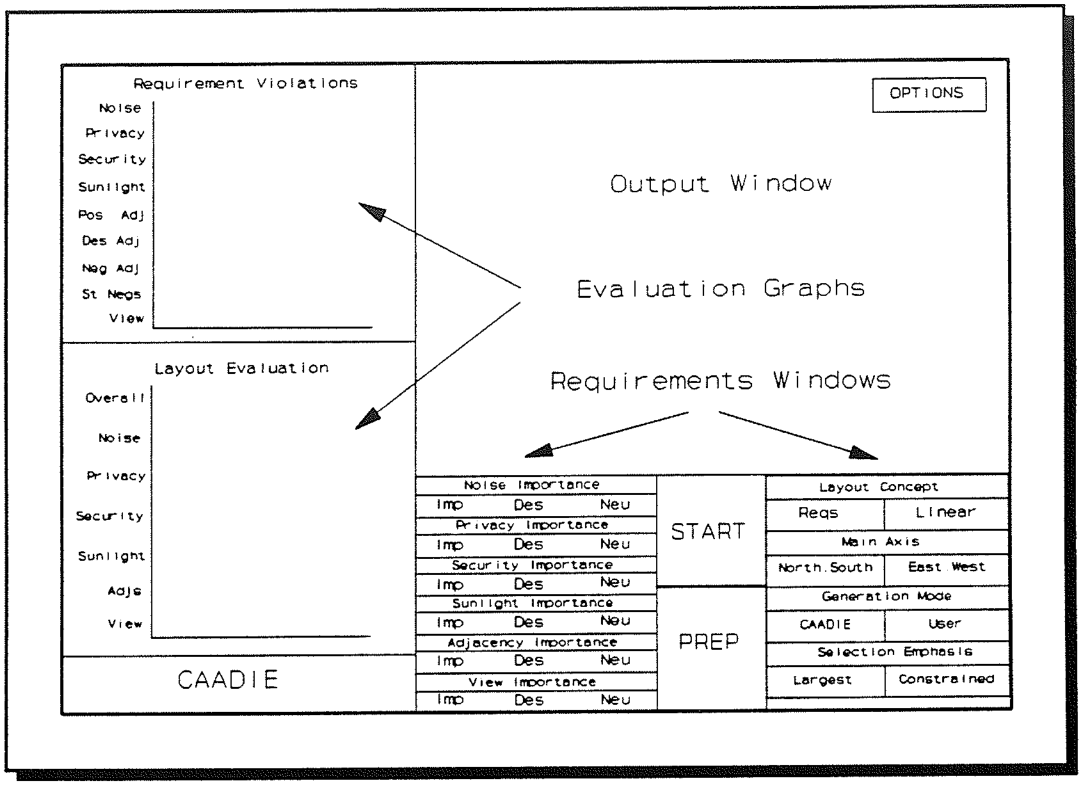


Figure 7-1: The CAADIE user interface

The following sections introduce output window and evaluation chart examples. The general requirement windows were previously introduced in relation to requirement generation, and thus, will not be reintroduced at this point. However, the graphic representation of layout requirements will be introduced to illustrate the potential of using graphic representations as an additional tool to communicate attribute values.

Output Window Results

The output window provides graphic representations of generated spatial relationship configurations. Figure 7-2 illustrates a simple, acoustics-based configuration as it is presented to the designer. The representation utilizes bubble diagram idioms to present spatial relationships created during layout generation. For example, the faculty office labelled FAC1 has been placed adjacent to the faculty office labelled FAC2. Additionally, the bubbles illustrate the relative square footage requirements of each space, and the total area requirement of the layout in relation to the allowable buildable area (represented by the rectangle encompassing the configuration).

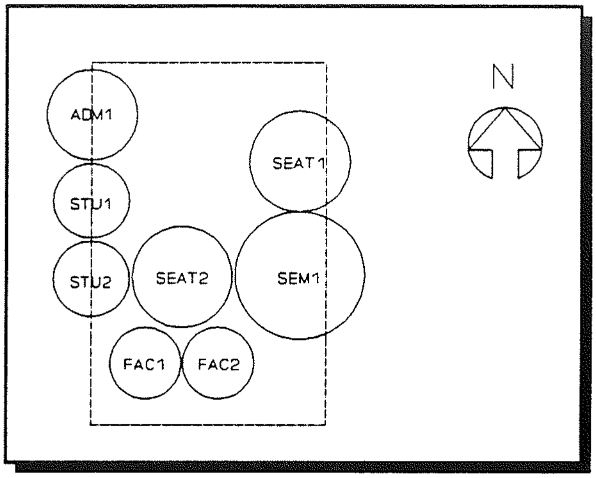


Figure 7-2: A simple bubble diagram based on acoustic zones

In addition to the ability to analyze relationships within a diagram, this idiom provides a convenient method to compare multiple diagram configurations. In the same manner as designers compare multiple alternatives by alluding to the placement of spaces within conceptual design sketches, designers may compare CAADIE-generated alternatives. Figure 7-3 depicts the same set of spaces regenerated with privacy selected as the focus attribute. A cursory comparison provides feedback concerning the effects of selecting the respective focus attributes. For example, the acoustics-based configuration is distinguished by its U-shaped form. Similarly, a closer comparison provides feedback

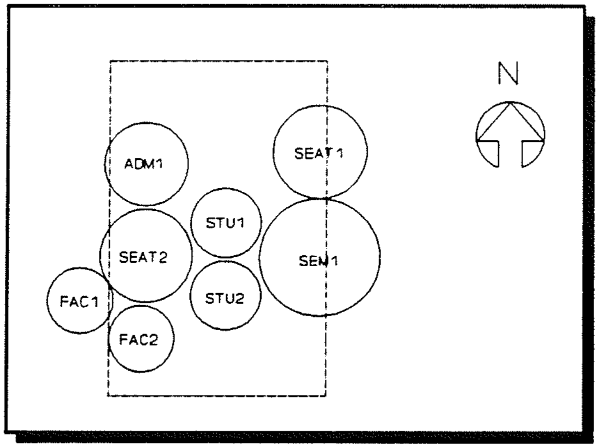


Figure 7-3: A simple bubble diagram based on privacy zones

concerning the impact of focus attributes on individual requirements such as external access and spatial adjacencies.

Graphic Requirement Representation

The influence of individual design attributes becomes obscured in the complex set of requirement trade-offs represented by final configurations. These trade-offs are often based on several factors including attribute importance, placement order, and design concepts. At times, the obscuring effect is desirable to focus attention on overall configuration characteristics such as layout compactness or design concept reinforcement. However, this effect is a detriment to examining the influence of individual attribute requirements on configuration relationships. To understand these influences, design attribute requirements must be isolated in the context of the proposed layout solution. The isolation of these attributes is the focus of the requirement representation options available in the CAADIE user interface.

Figure 7-4 illustrates the potential to assist designers in comprehending the effects of design attribute requirements by graphically isolating specific attributes. This figure represents daylighting requirements for the spaces within a proposed configuration. Similarly, view requirements and internal adjacency requirements may be isolated with graphic requirement representations (figures 7-5 and 7-6).

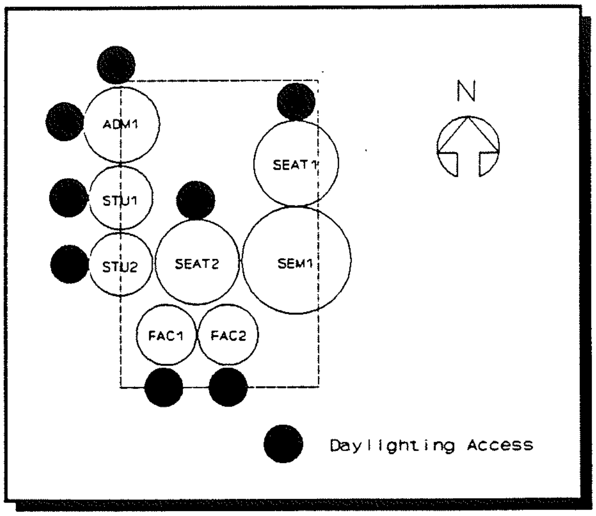


Figure 7-4: The graphic representation of daylighting requirements

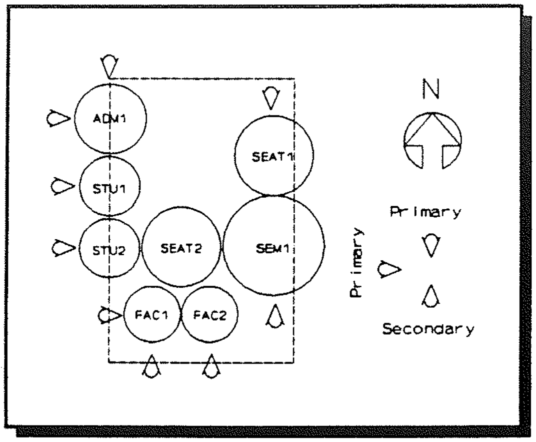


Figure 7-5: The graphic representation of view requirements

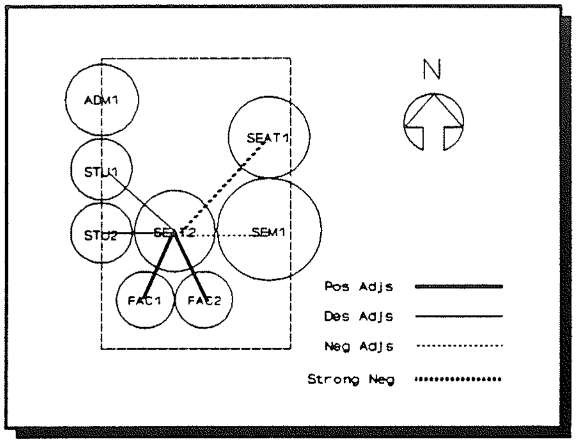


Figure 7-6: The graphic representation of adjacency requirements

In contrast to these adjacency-based requirements, security, acoustic, and privacy attributes do not explicitly require a space to be placed in a position relative to another space or external resource. To illustrate these relationships, the interface employs interior zone displays and attribute violation displays to isolate the effects of non-adjacency relationships. Zone displays highlight spaces in terms of their focus attribute requirements. For example, figure 7-7 depicts zone displays indicating the location of spaces according to their security requirements. The second form of display, the attribute violation display, indicates attribute requirements by highlighting incompatibilities between adjacent spaces. For example, figure 7-8 illustrates privacy incompatibilities detected within the given layout solution. Upon obtaining this information, a designer can elect to increase or decrease privacy importance factors to influence the trade-off of privacy requirements during the layout generation process.

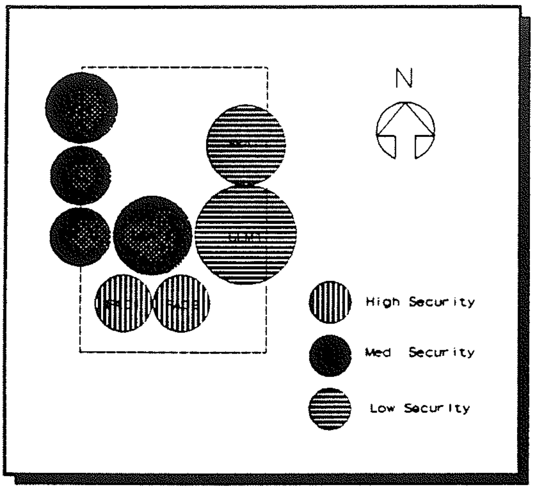


Figure 7-7: Interior zone display portraying security requirements

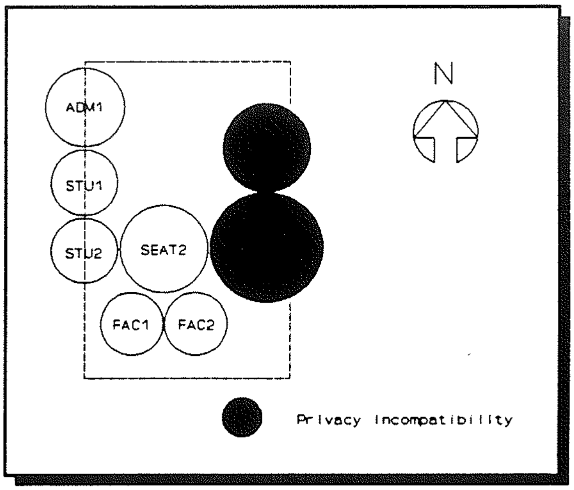


Figure 7-8: Portrayal of privacy requirement incompatibility

Evaluation Charts

The evaluation charts complete the design process emphasis by generating evaluation feedback. The CAADIE emphasis on attribute compliance requires an evaluation feedback medium to graphically convey both attribute compliance ratings and overall layout ratings. The graphic paradigm selected to accomplish this representation task is the histogram. This form of representation supports the ability to separate an overall layout rating into a compilation of individual attribute compliance ratings through a series of scaled bars. Additionally, this scaled representation displays the relative influence of each attribute on the overall layout rating. Thus, the designer receives the dual benefit of attribute compliance ratings and relative attribute influences, within the context of a single evaluation graph.

The CAADIE user interface incorporates two versions of these evaluation charts: an attribute violation chart and a weighted evaluation chart. The former chart provides the total number of attribute violations detected in the current configuration. The objective is to provide a baseline indication of the layout rating in terms of attribute compliance. Given these numbers, various layout alternatives can be compared exclusively on their ability to achieve attribute compliance. The weighted evaluation chart complements this feedback by graphically representing the individual attribute, and overall layout, ratings calculated during the layout generation process. These numbers scale each compliance rating in relation to the attribute importance factors and the total number of layout requirements. These ratings indicate which attributes contain the greatest impact on the layout rating in terms of both the number of attributes and the attribute importance. Additionally, each rating is presented together with its potential maximum value to indicate the degree to which attributes reach their maximum values. These differences can guide the designer in determining which attributes should be given more or less priority during the next design iteration.

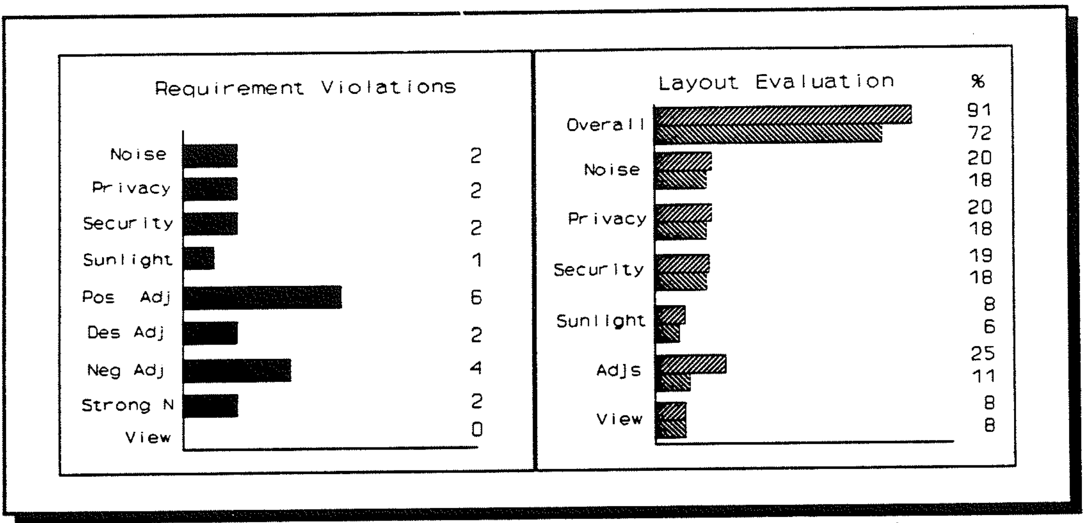


Figure 7-9: Evaluation ratings of the acoustics-based configuration

Figure 7-9 illustrates the evaluation chart implementation. The ratings in this example reflect the attribute compliance of the configuration illustrated in figure 7-2. As a comparison, figure 7-10 illustrates the evaluation ratings of the configuration portrayed in figure 7-3. An analysis of these graphs provides comparisons of the

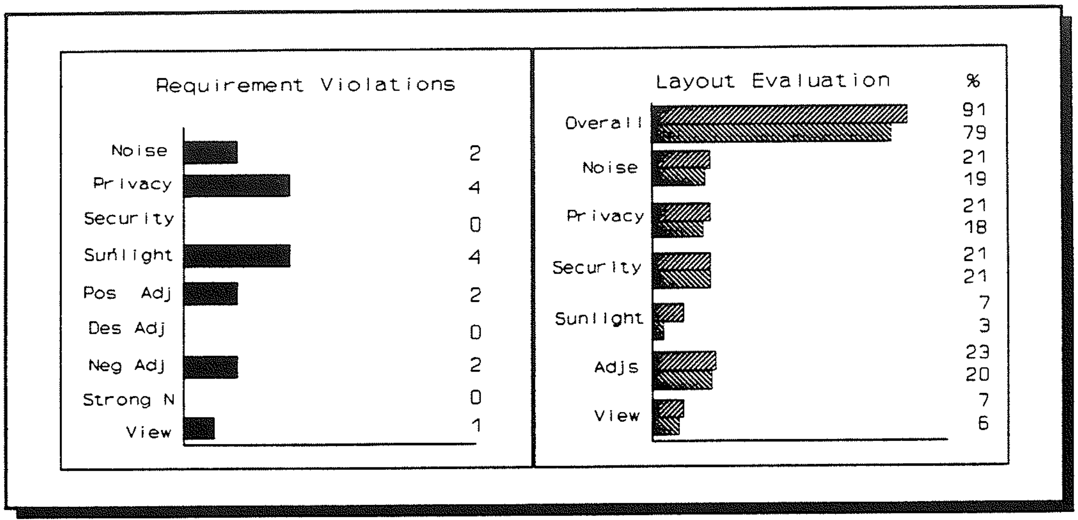


Figure 7-10: Evaluation ratings of the privacy-based configuration

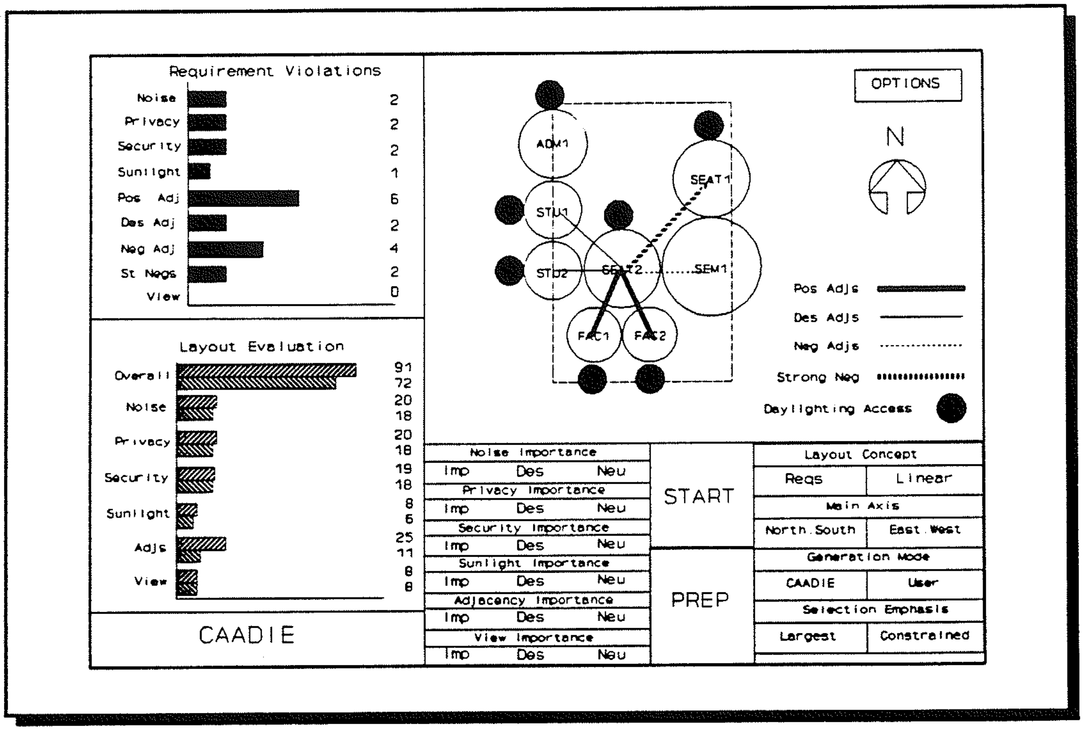


Figure 7-11: The CAADIE user interface with graphic elements

compliance trade-offs made to reinforce the respective focus attribute selections. When combined with the bubble diagram output, selected graphic requirement representations, and the general layout preferences, the resulting user interface appears as illustrated in figure 7-11.

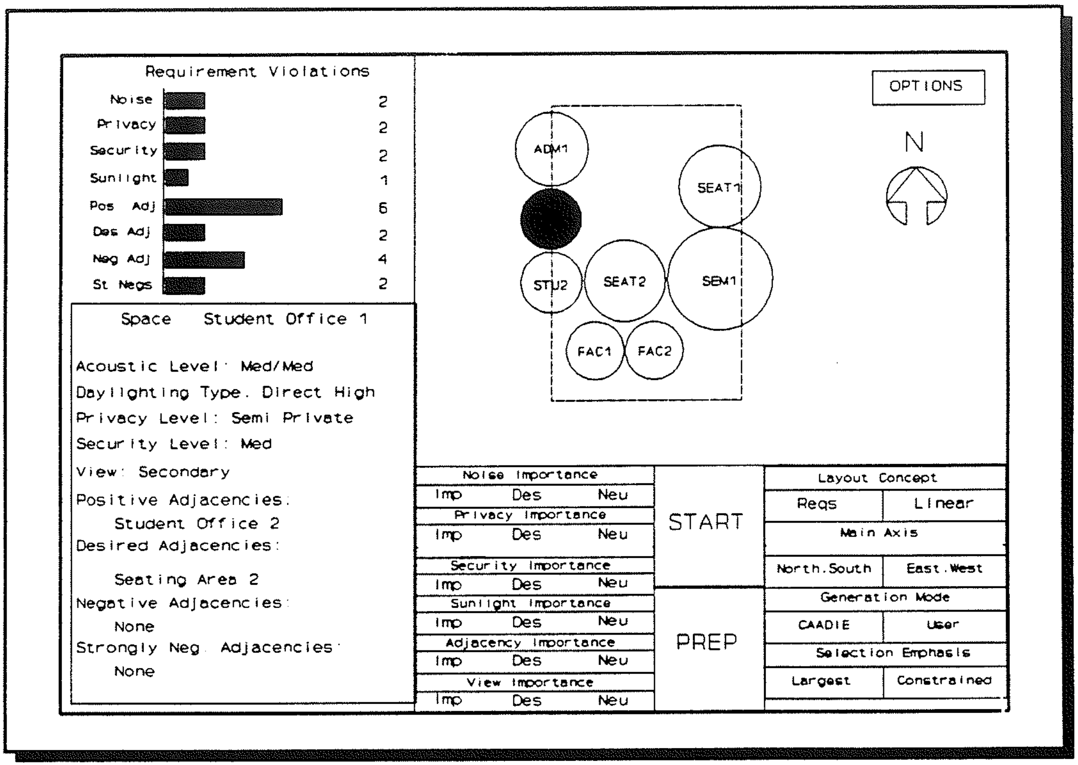
Layout Manipulation

The participatory design concept underlying the CAADIE project contains an additional impact in terms of the multiple configuration issue. As discussed previously, CAADIE emphasizes the generation of a single layout based on design principles, designer preferences, and typical layout requirements. However, this solution is not intended to be the final system result. Rather, the layout serves as a starting point from which to explore and refine configuration ideas through multiple layout configurations. To support this idea generation process, the user interface provides two layout manipulation capabilities: direct manipulation and attribute manipulation.

Direct Manipulation

Direct manipulation is the ability to select a space and manually move it to another location. This manual placement permits either a complete alteration of an existing configuration, or minor spatial location adjustments. In both cases, the system returns to the electronic sketch pad role first discussed in relation to the user initiated generation mode. In this role, CAADIE permits the designer to control layout development through an iterative process of altering spatial locations and receiving evaluation feedback.

The direct manipulation of spatial locations occurs through user interaction. As illustrated in figure 7-12, a space is selected for manipulation and the associated attribute requirements are displayed. The space may then be placed in any location adjacent to, or overlapping, the current configuration. If the space is adjoining another space, the configuration is retained as augmented. However, if the space is



Space Student Office 1

Acoustic Level: Med/Med
Daylighting Type. Direct High
Privacy Level: Semi Private
Security Level: Med
View: Secondary
Positive Adjacencies.
    Student Office 2
Desired Adjacencies:
    Seating Area 2
Negative Adjacencies:
    None
Strongly Neg. Adjacencies:
    None

Figure 7-12: The user interface displays spatial requirements when a space is selected for manipulation

placed in a position which overlaps previously placed spaces, then the configuration is modified according to the least movement algorithm. Upon completing this alteration, CAADIE updates the evaluation charts to indicate the current attribute compliance ratings (figure 7-13). Based on this feedback, the designer may elect to continue the alteration process until an acceptable configuration is achieved.

The CAADIE system provides an additional level of flexibility for developing multiple configurations by permitting layout variations to be created from any layout previously generated in the current session. These variations may occur incrementally, or in a single layout alteration. Each of these variations will be stored for future reference. Subsequently, the designer may toggle between alternatives to pursue different layout options. This lends an added degree of similarity to the

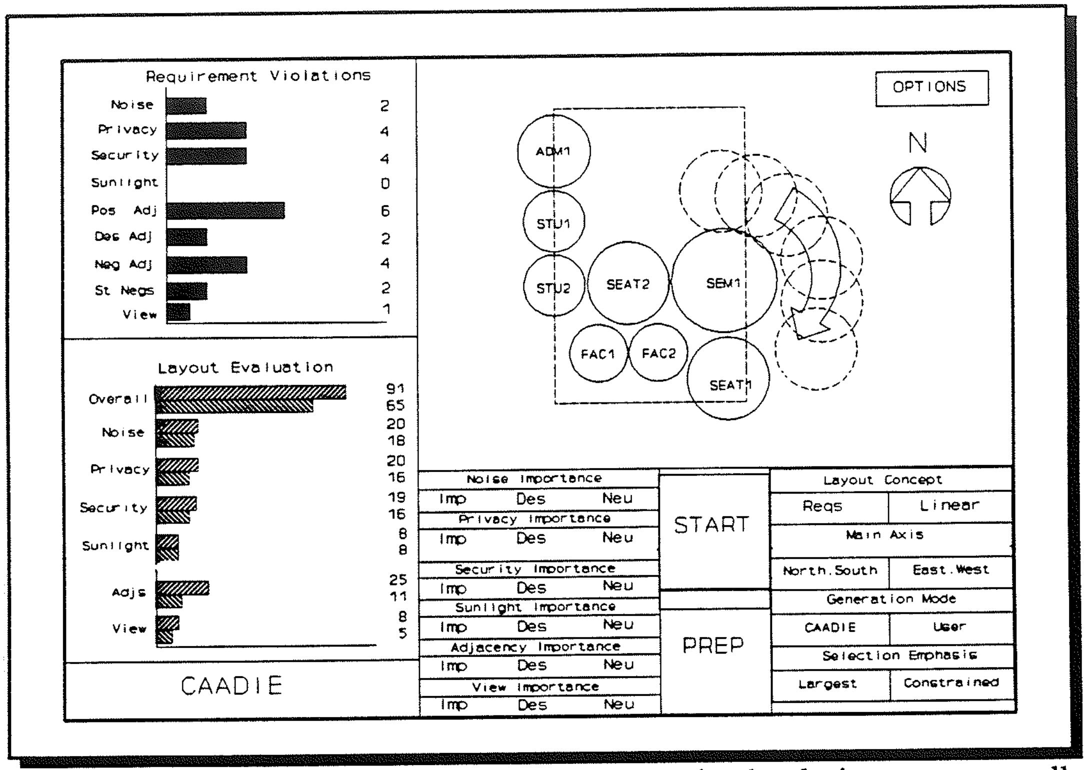


Figure 7-13: Direct manipulation of a space permits the designer to manually alter spatial locations and receive updated compliance feedback

designer's manual process of generating multiple configurations in the pursuit of a final layout solution.

Attribute Manipulation

Attribute manipulation supplements direct manipulation by providing the ability to focus on individual attributes affecting a design object. This manipulation represents an opportunity to perform "what-if" types of analyses in association with design attributes. In these analyses, a designer may alter an attribute requirement, or group of attribute requirements, for any design object, to examine the impact of the change on the overall configuration. Similarly, individual attribute importance ratings may be altered at either the overall or design object level to explore the impact of this alteration. For example, the daylighting requirements of several spaces, or the overall daylighting importance factor, may be altered to examine the impact of daylighting on layout configurations.

In the same manner as direct manipulation occurs, attribute manipulations may be performed over a series of incremental stages. A designer can elect to create several layouts, each of which incorporates a single attribute manipulation, to gain a greater understanding of design attribute influence. This process is especially useful for refining the design program which will be used as a basis for the remaining layout generation stages. It may also be a useful approach when working with a client to help evaluate alternative client priorities.

Future Implications

The CAADIE user interface achieves the goal of supporting an interactive approach to layout generation. However, the CAADIE prototype addresses only a small segment of the layout generation process. In future system implementations, the scope of the generation process would be expanded to include additional design process stages and additional design attributes. This expansion creates ramifications for the user interface in terms of the capability to accommodate the requirements of these additional stages and attributes.

- *What are the requirements of additional layout generation stages?* - The current emphasis on conceptual diagrams is sufficiently accounted for within the user interface capabilities. However, the requirements of successive layout generation stages are significantly more complex in terms of layout representation. Whereas bubble diagrams have considerable flexibility, subsequent architectural diagrams require accurate graphic representations to realistically portray the dimensional aspects of each design object. Accommodating these requirements has traditionally occurred through the use of computer-aided drafting systems. Thus, the issue arises as to the appropriate environment for progressing from conceptual design stages to detailed design stages.

Options for addressing this progression include integrating knowledge-based design systems into existing drafting systems, developing drafting capabilities in knowledge-based design systems, and transferring data between drafting systems and conceptual design systems. Although a final solution to this user interface issue is beyond the scope of the current CAADIE project, concurrent research efforts indicate that integrating knowledge-based capabilities into drawing systems appears to be a potential solution (Ito, Law, and Levitt 1990). Given the relative requirements and advantages of each option, this approach has proven to be the most effective at this time. However, additional study will be required to satisfactorily achieve a solution to this multi-stage issue.

- *What are the requirements of additional design attributes* - Design attribute representation is currently addressed by the graphic options within the user interface. However, the current system is limited to addressing the requirements of the six design attributes incorporated in the CAADIE knowledge base. In future system implementations, this range of attributes would be expanded to include issues as diverse as handicapped access and fire code compliance. In contrast to the multi-stage issue, limited study has been undertaken in regards to representing these multiple attributes within a graphic interface. Based on the importance of design attribute representation, the development of an expanded design system will necessitate a study of these additional attribute requirements as a fundamental project objective.

CHAPTER 8

CONTRIBUTIONS AND FUTURE WORK

Contributions to Knowledge

The theoretical research and implementation studies conducted within the CAADIE project achieve the objectives outlined in Chapter 1. In achieving these objectives, the research advances the following contributions to knowledge.

Demonstrates the viability of, and requirement for, a participatory design concept

Chapter 4 presented the CAADIE approach to layout generation emphasizing a participatory design environment in which designers retain the prerogative to determine appropriate assistance levels and layout generation processes. Through this interactive environment, designers impart design preferences which reflect individual experiences and intuition. Subsequently, this experiential knowledge complements the established design principles within the system, by serving as guidelines for the adaptation of layouts to the current problem requirements.

Additionally, the participatory design concept provides the ability to explore layout issues through an electronic sketch pad analogy. This manual design mode creates an iterative design session in which designers retain the capability to generate layout alternatives while CAADIE provides evaluation feedback. This capability emphasizes the system's role as a design assistant by returning layout generation control to the designers. It is this flexibility to alter the degree of automation, and amount of knowledge imparted by designers, which transforms CAADIE from an automated design generator to a design assistant. The successful demonstration of this concept in the CAADIE prototype should create a new emphasis in layout generation research away from the development of automated systems to the development of participatory design environments.

Introduces a framework for modelling the information the information and knowledge used by designers during the layout generation process.

Chapter 5 introduced the CAADIE knowledge model as a framework for capturing the spectrum of layout information and knowledge impacting layout configurations. Within this overall framework, the model presents a series of categories identifying the layout information and design expertise areas required to address layout generation issues. The knowledge model enhances layout generation in two important areas: 1) the storage of typical layout requirements, and 2) balancing the overall layout and space-specific requirements.

The storage of typical layout requirements reduces reliance on designer input, and capitalizes upon the use of typical information within similar layout problems. This increases the designer's ability to focus on design activities, rather than information input activities. Concurrently, this decreases the disparity between the degree of information imparted to the system and the layout generation assistance returned. In the future, this will increase designer acceptance, and provide a basis for introducing layout generation systems into the design profession.

Balancing the need for design program compliance with the necessity for underlying design rationale bridges the gap between satisficing systems and design methodology systems. The combination of design concept heuristics and typical design object requirements results in layout decisions which reinforce selected concepts, while retaining a grounding in design program reality. This layout generation premise lies at the center of the CAADIE philosophy, and represents one of the primary requirements given by the designers involved with the CAADIE research effort. It is intended that the facilitation of this premise by the proposed knowledge model will continue to promote a balance between layout issues within future layout generation research efforts.

Demonstrates the potential for combining knowledge-based paradigms with information processing and graphic capabilities to address a greater range of layout generation issues.

The layout generation issues enumerated in Chapter 2 present a baseline definition of the design and computational factors impacting layout generation research. The CAADIE approach to addressing these issues builds upon concepts introduced by several researchers including Akin, Pohl, and Stiny. Within this approach, knowledge-based paradigms are utilized to overcome the limitations of algorithmic-based systems, and to address a greater number of design issues. Specifically, knowledge-based paradigms capture designer strengths such as high-level decision making and information organization, to supplement the information processing and graphic representation capabilities found in computers. In this way, the ability to address multiple design requirements is retained together with high-level decision making capabilities.

The CAADIE prototype validates this approach by addressing a spectrum of issues from design knowledge to design concepts. Moreover, the prototype presents a framework which **does not** preclude the remaining layout generation issues from being included in the CAADIE system. The system architecture permits any domain to be incorporated through the use of modular rule sets. In contrast, algorithmic-based systems are restricted from addressing the complete range of layout generation issues due to the inability to include design knowledge within their base formalisms. Therefore, the CAADIE results put forth the argument that an architecture based on knowledge-based representation and reasoning paradigms, supplemented by information processing and graphic procedures, is essential as a layout generation basis.

Demonstrates the importance of an iterative analysis-synthesis-evaluation cycle as a basis for layout generation.

Chapters 2 and 6 highlighted the central role an iterative design process plays in the generation of design solutions. Designers iterate between generating design requirements, synthesizing configurations, and evaluating solutions, in an effort to explore and understand individual and general design issues. Disregarding this process, as was done in initial layout generation research, eliminates the potential to provide designers with valuable evaluation feedback. To emphasize this feedback capability, the iterative analysis-synthesis-evaluation cycle forms the central design process within the CAADIE system.

The implementation of this iterative process emphasizes the ability to build upon evaluation feedback by providing layout manipulation and user-initiated generation capabilities. Given evaluation feedback based on objective factors, the designer is able to alter layout requirements and spatial locations in accordance with the evolving solution. Additionally, this feedback provides a basis from which to compare multiple layout alternatives. Although CAADIE implements this iterative process within a single design phase, evaluation and manipulation are fundamental design components, and will provide an ever increasing amount of feedback as layout refinement proceeds.

Identifies and supports the designer's viewpoint

The knowledge acquisition process outlined in Chapter 4 furnished the knowledge and information necessary to develop and implement the CAADIE knowledge model. Additionally, the process provided an insight into the viewpoint from which architectural designers approach the layout generation problem. The development of the CAADIE prototype around this viewpoint diverges from concurrent efforts by incorporating an approach emphasizing flexibility and adaptation. This flexibility acknowledges designer tendencies to iterate between design process phases and multiple design stages, while concurrently adapting previous experience to the present layout instance.

The resulting impact of this emphasis is evidenced by the options available in the CAADIE prototype (i.e., multiple generation modes, overall requirement preferences, and manipulation capabilities). Based on the positive response to these options, and the potential use of these options as central design aids, the conclusion has been reached that this viewpoint will be an essential component of commercial CAAD systems, and should represent a fundamental objective of future layout generation research efforts.

CAADIE Limitations

The CAADIE prototype serves as a proof of concept for the CAADIE research effort. As a proof of concept system, it contains limitations in several areas impacting configuration development. The following sections discuss some of these issues and possible ways to overcome the limitations.

Partial Configuration Generation

The CAADIE solution methodology centers on aggregating individual design decisions to achieve overall configurations based on established design principles. This methodology retains the advantage of reducing the number of partial configurations existing within the system. Concurrently, the disadvantage is retained which requires all layout decisions to be made at a local level (i.e., when the space is being placed). CAADIE does not permit placement decisions to be postponed until later in the problem solving process. Therefore, the possibility arises that a local decision may be preferred at that point in the layout generation process, but in the overall configuration context, a less preferred option may have resulted in a higher rated layout.

Chapter 6 outlined the reasons for selecting this procedure including an elimination of computational explosion possibilities and a reduction in analysis requirements. However, CAADIE should incorporate additional options for eliminating partial configurations. For example, one possible approach would be to include a fuzzy logic component which evaluates placement options as relative degrees away from the most preferred placement option. In this way, the designer could be given the flexibility to set the threshold level which placement options must surpass to be considered acceptable. All options exceeding this threshold would be retained as partial configuration alternatives, and extended through the addition of further spaces. At one extreme, the selection of a stringent threshold would result in a single option being accepted at any given time (as is the current situation), while at the other extreme, all eligible options would be retained for further development.

This example represents one approach to increasing the number of partial configurations pursued during the layout generation process. Although many other options exist, the underlying problem of cost and return remains a concern for each option. Specifically, the benefit of pursuing these alternatives must be determined to outweigh the costs of retaining the alternatives in order for these methods to be an improvement over the current CAADIE implementation.

Two-Dimensional Problem Solution

The overwhelming majority of systems developed over the past three decades address the layout generation problem on a two-dimensional scale. The decision was made early in this research effort to similarly restrict the CAADIE prototype. The underlying reason for this decision centered on the additional level of reasoning required to generate and evaluate placement option decisions. The above and below relationships evident in three-dimensional problems represent a notable increase in reasoning complexity. These relationships impact many layout requirements including adjacencies, lighting, and circulation. Additionally, overall building concerns such as three-dimensional form, become primary factors in evaluating both individual placement options and overall layout configurations.

The decision to implement CAADIE as a two-dimensional problem solver is not intended to be a permanent restriction. The underlying architecture does not preclude three-dimensional reasoning from being incorporated into the decision making processes. However, the current lack of three-dimensional reasoning prevents this research effort from making any definitive claims as to the difficulties or hindrances associated with adding this reasoning capability. It is projected that this addition is possible within the current framework, but future research efforts will be required to substantiate this claim.

Explanation and Design Histories

The constraint representation options available in the CAADIE user interface provide access to the underlying reasons behind layout generation decisions. The use of constraint representations yields an insight into the configuration rationale, but are limited in their capability to provide complete design histories. Notably, the spatial ordering heuristics and layout requirements used to guide individual placement decisions are not available in a comprehensive format. The addition of design histories could provide valuable information for both exploring alternative layout options, and reviewing design decisions made during earlier layout generation stages.

The implications of requiring additional explanation capabilities within a computer-aided design system are not clear at this time. However, initial research is currently being conducted to create appropriate design history models (Garcia and Howard 1991). The incorporation of these models could eventually enhance the

CAADIE explanation capabilities, and bring designers closer to the decision making processes.

Future Work

The CAADIE system introduces several contributions to the layout generation research domain. Based on these contributions, several areas emerge as potential commencement points for expanding the contributions in future research efforts. First, the CAADIE research could be *extended* to address a greater number of layout generation issues and design domains. Second, future research could *focus* on issues briefly introduced in the CAADIE system such as appropriate user interfaces. Finally, *integrating* the CAADIE prototype within a larger design system context is a potential research area. The following sections provide a further examination of these potential future research efforts.

Project Extensions

Extending the CAADIE project beyond its current scope emphasizes two specific areas:

- Additional design attributes
- Additional process stages

The addition of further design attributes beyond the six currently addressed, will require corresponding frame hierarchy expansions and rule set expansions. In terms of the frame hierarchies, additional design attribute classes and subclasses will be required to accommodate the expanded layout information associated with each attribute. Concurrently, additional rule sets containing analysis, synthesis, and evaluation heuristics will be required to capture the appropriate design attribute expertise. In respect to several design attributes including HVAC and handicapped requirements, this will not be a trivial task. Protocol studies investigating the roles these attributes play in the layout generation process will be needed to ensure that the same level of decision making integration is attained as is evident with the existing design factors. Finally, conflict resolution and knowledge selection heuristics will require modifications to accommodate the expanded range of design attribute focus areas.

The addition of multiple design stages represents the greatest system alteration. As previously discussed, the multiple stage issue is a primary indicator of design system capabilities. The number of stages addressed determines the design process spectrum in which the system is capable of providing design assistance. Expanding this range requires additional rule sets to perform design tasks such as transforming geometric shapes, generating detailed design decisions, and evaluating various design abstraction levels. Moreover, this knowledge will be required to address individual design attributes. For example, progressing from conceptual to detailed design stages in terms of daylighting, requires an extension beyond external access concerns to knowledge focusing upon issues such as interior light levels, window sizes, and window locations.

The uniqueness of the knowledge associated with each attribute and design process stage has not been addressed in this research, however, it is anticipated that each of these areas will encompass individual knowledge requirements. Thus, expanding CAADIE to address either additional attributes or additional stages could potentially provide a series of individual research efforts.

Project Focus Areas

The second research area focus is associated with furthering research in relation to specific CAADIE prototype issues. The greatest opportunity in this area is related to future user interface development. The KEE platform provided sufficient graphic capabilities from which to create a participatory design environment. However, the potential to expand the user interface is apparent in several areas including greater drawing capabilities, enhanced knowledge access, and the addition of a learning component.

The first of these areas, a greater drawing capability, could be addressed by combining CAADIE layout generation capabilities with a commercial CAD system's drawing capabilities. This integration would provide layout stages requiring greater accuracy than bubble diagrams with advanced grid and scaling options. Additional enhancements would be applicable to the graphic requirement representations, manipulation options, and evaluation graphs. Each of these areas is dependent upon the maximum use of natural idioms and tools to effectively interact with designers. Increasing the capability to create these idioms will correspondingly increase designer participation.

The second area, expanded knowledge access, was previously addressed in Chapter 7. In brief, the ability to access the underlying knowledge model provides designers with the opportunity to customize the knowledge to individual preferences. The current CAADIE prototype limits knowledge model access to KEE-based editing procedures. This process is inappropriate for everyday designer use. A research opportunity exists to create an appropriate front-end to this process which would render the underlying KEE editing facilities transparent to the designer.

Finally, the learning component represents the greatest research challenge. The participatory environment espoused in this project entails extensive designer interaction. Designer initiated actions are associated with a series of goals established to investigate specific design issues. For example, when a designer alters previously determined spatial locations, the underlying goal may be to create additional external access pathways. It is this underlying goal which is the learning component focus. In the context of limited goals such as the previous example, it should be possible to create heuristics identifying conditions under which additional layout requirements could be inferred or learned. For example, the following heuristic represents a learning component for identifying external access requirements.

IF A space had an external access direction previously blocked
AND The direction is now accessible due to layout manipulation
THEN
Add an external access requirement to the space in the previously blocked direction

Successfully implementing this component would require extensive protocol analysis studies to identify the underlying goals driving designer actions. However, through initial efforts focused upon limited domains such as external access requirements, notable advancements are possible. Given these preliminary achievements, expanded learning capabilities could be added throughout the layout generation process.

CAADIE Integration

The final research opportunity addresses the integrated design system objective. Chapter 3 introduced several efforts underway to create integrated design environments. These integrated environments incorporate several individual systems to collectively generate design solutions. Within this integrated environment, it is foreseeable that CAADIE could provide one element in an overall design system. CAADIE layouts could be used as input by structural and environmental systems for the development of appropriate building support systems. Subsequently, design information could be provided to construction and estimating systems for further design development studies.

The essential research component associated with this integration is the data transfer problem. The CAADIE knowledge model is appropriate for layout generation activities. However, the appropriateness of this model for additional design systems is questionable. In all probability, each system will require its own data viewpoint. Thus, determining how to transfer data to and from the knowledge model becomes a central research question. The solution to this issue will provide the first steps towards incorporating CAADIE within an overall design system which provides assistance from initial design program development to facility management.

APPENDIX A

CAADIE SPACE DEFINITION FILE

Generic Definition

Definition File Entry	Description
create.unit	KEE command to create frame instance
'faculty.office1	The name for the new space
'thesis1	The name of the knowledge base in which the frame will be created
nil	A place holder indicating that this frame is not a new subclass definition
'(faculty.offices)	A list of subclasses to which the frame will be attached as a child
nil	A place holder indicating that this frame does not contain new member slot definitions or values
nil	A place holder indicating that this frame does not contain new local slot definitions or values

Space Definitions

```
(create.unit 'faculty.office1
              'thesis1
              nil
              '(faculty.offices daylighting.faculty.offices human.factors.faculty.offices
space.planning.faculty.offices)
              nil
              nil)

(create.unit 'faculty.office2
              'thesis1
              nil
              '(faculty.offices daylighting.faculty.offices human.factors.faculty.offices
space.planning.faculty.offices)
              nil
              nil)

(create.unit 'faculty.office3
              'thesis1
              nil
              '(faculty.offices daylighting.faculty.offices human.factors.faculty.offices
space.planning.faculty.offices)
              nil
              nil)

(create.unit 'faculty.office4
              'thesis1
              nil
              '(faculty.offices daylighting.faculty.offices human.factors.faculty.offices
space.planning.faculty.offices)
              nil
              nil)

(create.unit 'student.office1
              'thesis1
              nil
              '(student.officesdaylighting.student.officeshuman.factors.student.offices
space.planning.student.offices)
              nil
              nil)
```

```
(create.unit 'student.office2
                'thesis1
                nil
                '(student.officesdaylighting.student.officeshuman.factors.student.offices
space.planning.student.offices)
                nil
                nil)

(create.unit 'seating.area1
                'thesis1
                nil
                '(seating.areas   daylighting.seating.areas   human.factors.seating.areas
space.planning.seating.areas)
                nil
                nil)

(create.unit 'seating.area2
                'thesis1
                nil
                '(seating.areas   daylighting.seating.areas   human.factors.seating.areas
space.planning.seating.areas)
                nil
                nil)

(create.unit 'seminar.room1
                'thesis1
                nil
                '(seminar.rooms daylighting.seminar.rooms human.factors.seminar.rooms
space.planning.seminar.rooms)
                nil
                nil)

(create.unit 'seminar.room2
                'thesis1
                nil
                '(seminar.rooms daylighting.seminar.rooms human.factors.seminar.rooms
space.planning.seminar.rooms)
                nil
                nil)
```

```
(create.unit 'administrative.office1
             'thesis1
             nil
             '(administrative.offices daylighting.administrative.offices
human.factors.administrative.offices space.planning.administrative.offices)
             nil
             nil)

(create.unit 'general.classroom1
             'thesis1
             nil
             '(general.classrooms daylighting.general.classrooms
human.factors.general.classrooms space.planning.general.classrooms)
             nil
             nil)

(create.unit 'restroom1
             'thesis1
             nil
             '(restrooms daylighting.restrooms human.factors.restrooms
space.planning.restrooms)
             nil
             nil)

(create.unit 'restroom2
             'thesis1
             nil
             '(restrooms daylighting.restrooms human.factors.restrooms
space.planning.restrooms)
             nil
             nil)

(create.unit 'research.lab1
             'thesis1
             nil
             '(research.labs daylighting.research.labs human.factors.research.labs
space.planning.research.labs)
             nil
             nil)
```

APPENDIX B

EXAMPLE CAADIE SESSION

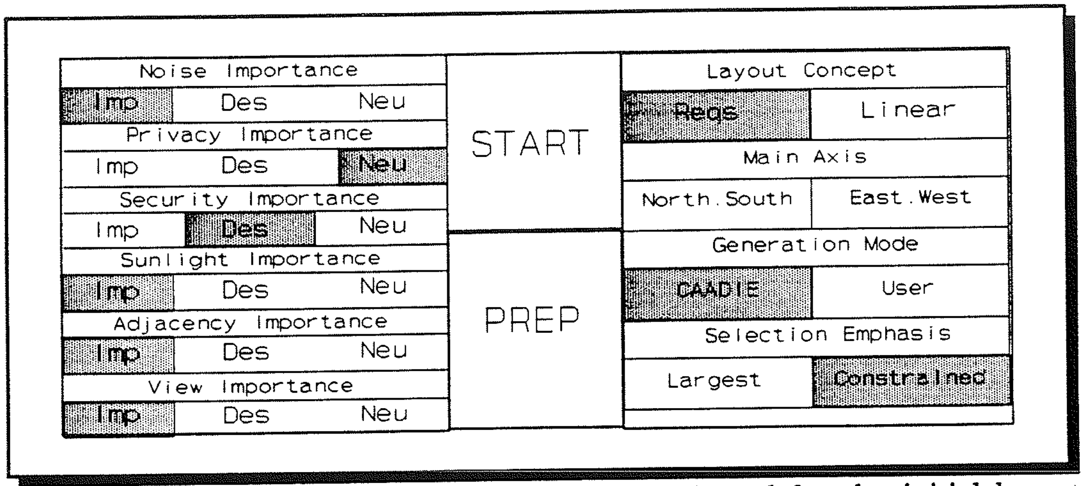
The following scenarios provide a representative CAADIE design session example. The scenarios highlight five primary layout generation capabilities facilitated by the cooperative design environment:

- CAADIE Layout Generation
- Direct Manipulation
- Attribute Manipulation
- Linear Concept Generation
- User Initiated Layout Generation

Space Definition and Analysis

The definition file given in Appendix A defines the spaces which will be placed in the following scenarios. The fifteen spaces represent a subset of the spaces typically included in university research buildings (International Council of Educational Facility Planners 1985). The combination of these spaces presents CAADIE with a realistic layout generation problem based on education committee recommendations (Castaldi 1987).

The space definition file initializes the system to begin the analysis-synthesis-evaluation process. Figure B-1 illustrates the general layout requirements selected for the initial layout generation session. Additionally, *acoustics* has been selected as the focus attribute, and the buildable area is restricted to an 85x60 area with a 50% overlap variance. This variance accommodates the extra area required by the amorphous shapes. In later stages, this variance will be reduced in accordance with dimension and shape refinement.



Noise Importance
[✓ Imp] Des Neu
Privacy Importance
Imp Des [✓ Neu]
Security Importance
Imp [✓ Des] Neu
Sunlight Importance
[✓ Imp] Des Neu
Adjacency Importance
[✓ Imp] Des Neu
View Importance
[✓ Imp] Des Neu

Layout Concept
[✓ Reqs] Linear
Main Axis
North.South East.West
Generation Mode
[✓ CAADIE] User
Selection Emphasis
Largest [✓ Constrained]

Figure B-1: The general layout requirements selected for the initial layout generation session

Concurrent to obtaining general layout requirements, the analysis stage generates space-specific requirements. As discussed in Chapter 6 (see page 100), CAADIE generates a complete set of requirements based on the layout information stored in the knowledge model. Each space is instantiated as an instance of the appropriate frame hierarchy subclasses, and consequently, inherits the space-specific requirements. At the conclusion of this stage, each space contains the requirements and guidelines necessary to support synthesis tasks (figure B-2).

Scenario I - CAADIE Initiated Layout Generation

The selection of acoustics as the focus attribute provides the organization basis for developing internal zones. These zones are developed according to the relative rigidity of the focus attribute requirements (i.e., the zones with the most stringent requirements are developed prior to zones with less stringent requirements).

#	Space Name	Positive Adjacencies (Space No.)	Desired Adjacencies (Space No.)	Negative Adjacencies (Space No.)	Strongly Negative Adjacencies (Space No.)	Acoustics Tolerance/ Output
1	Admin. Office 1	9	---	---	---	Med/Med
2	Student Office 1	3, 8, 13 14	10	---	---	Med/Low
3	Student Office 2	2, 8, 13 14	10	---	---	Med/Low
4	Faculty Office 1	5, 6, 7, 8 10	---	---	15	Low/Low
5	Faculty Office 2	4, 6, 7, 8 10	---	---	15	Low/Low
6	Faculty Office 3	4, 5, 7, 8 10	---	---	15	Low/Low
7	Faculty Office 4	4, 5, 6, 8 10	---	---	15	Low/Low
8	Research Lab 1	2, 3, 4, 5 6, 7	---	---	11, 12	Med/Med
9	Seating Area 1	1	---	13, 14, 15	10	High/ High
10	Seating Area 2	4, 5, 6, 7	2, 3	13, 14 15	9 .	High/ Med
11	Rest-room 1	---	15	---	8	High/ Med
12	Rest-room 2	---	15	---	8	High/ Med
13	Seminar Room 1	2, 3	---	9, 10	---	Low/Med
14	Seminar Room 2	2,3	---	9, 10	---	Low/Med
15	Class-Room 1	---	11, 12	9, 10	4, 5, 6, 7	Low/ High

Figure B-2 (Part I): The space-specific requirements generated by the system for the fifteen spaces.

#	Space Name	Privacy Req't	Security Req't	Sunlight Access Option	View Access Option	Important Atts	Desired Atts	Neutral Atts
1	Admin. Office 1	Semi-Private	High	N S E W	N	View	Sunlight	---
2	Student Office 1	Private	Med	E W	N S W	---	View	---
3	Student Office 2	Private	Med	E W	N S W	---	View	---
4	Faculty Office 1	Private	High	S	S W	View, Sun Noise	Security	---
5	Faculty Office 2	Private	High	S	S W	View, Sun Noise	Security	---
6	Faculty Office 3	Private	High	S	S W	View, Sun Noise	Security	---
7	Faculty Office 4	Private	High	S	S W	View, Sun Noise	Security	---
8	Research Lab 1	Private	High	N	N S W	Privacy, Security	Noise	---
9	Seating Area 1	Public	Low	---	N	Privacy	Noise	Sun
10	Seating Area 2	Semi-Private	Med	N S E W	N	Privacy	Noise	Sun
11	Rest-room 1	Public	Low	---	---	Privacy	Adjacencies	Sun, View
12	Rest-room 2	Public	Low	---	---	Privacy	Adjacencies	Sun, View
13	Seminar Room 1	Public	Low	N	---	Noise	Security	Privacy
14	Seminar Room 2	Public	Low	N	---	Noise	Security	Privacy
15	Class-Room 1	Public	Med	N	---	Noise	Security	Privacy

Figure B-2 (Part II): The space-specific requirements generated by the system for the fifteen spaces

Based on this development strategy, and the CAADIE space selection procedure (see page 106), *faculty office 4* is selected as the first space for placement. Following this selection, configuring the low tolerance/low output acoustics zone becomes the first synthesis objective.

The subsequent selections and placements reinforce the internal zone development according to the space selection heuristics, spatial ordering heuristics, and layout requirements (figures B-3 and B-4). Similarly, the placement generation process progresses through additional zones by selecting new focus spaces which meet the conditions specified in the space selection heuristics (figure B-5). For example, *research lab 1* is selected as a focus space after *faculty office 4*. This selection is based on several factors including: *faculty office 4* containing no further positive or desired adjacencies, the completion of the low tolerance/low output zone, and *research lab 1* containing the greatest number of adjacency constraints.

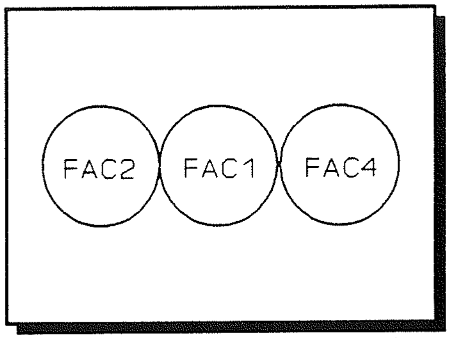


Figure B-3: The first selections and placements reinforce the low tolerance/low output acoustics zone

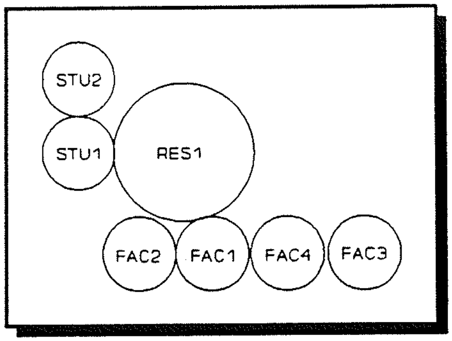


Figure B-4: The selections and placements continue to reinforce the internal zones

The emphasis on internal zones and spatial ordering concepts is retained throughout the layout generation process. However, a balance between this emphasis and the layout requirements is maintained through the evaluation and conflict resolution processes. Figure B-6 illustrates this balance in the context of *classroom 1*. The preferred placement option, location 1, is unavailable due to buildable area constraints. Similarly, the remaining placement options generated by the spatial ordering heuristics each violate layout requirements. In this circumstance, the conflict resolver relaxes individual requirements according to the conflict resolution heuristics (see page 113). This process continues until an option is generated which meets both the relaxed layout requirements and the spatial ordering concept preferences. In this example, the acoustics requirement of *classroom 1* and the view requirement of *seating area 2* have both been relaxed prior to the generation of a successful placement option (location 2).

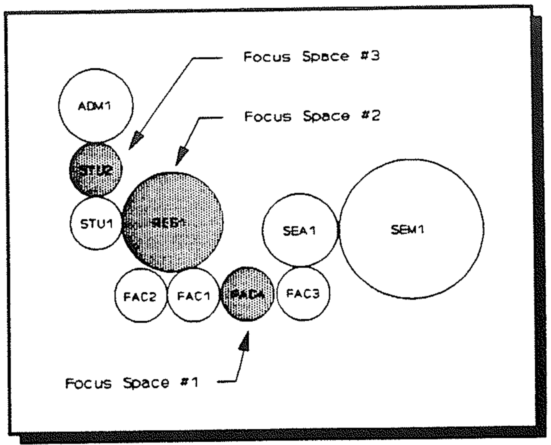


Figure B-5: The selection of focus spaces facilitates the development of internal zones

At the conclusion of the layout generation process, a complete layout is presented to the designer together with attribute compliance ratings (figure B-7). The evaluation charts indicate the number of attribute violations, the relative impact of each attribute on the overall rating, and the scaled overall rating. The designer may supplement this feedback with graphic requirement and attribute violation displays (figures B-8 and B-9).

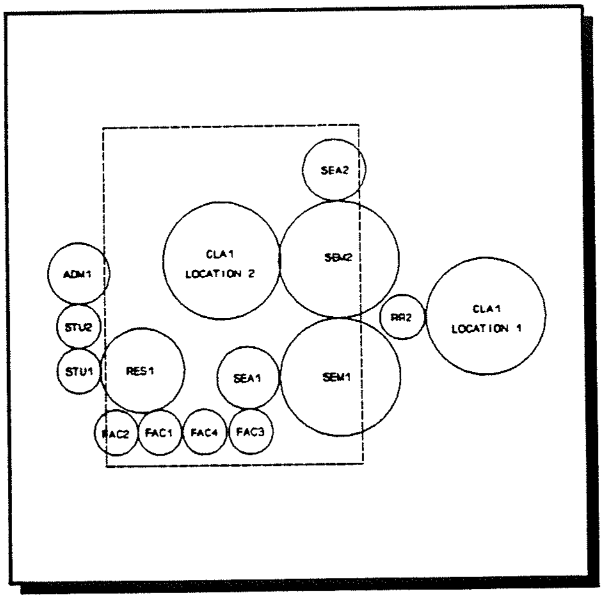


Figure B-6: A balance is retained between concepts and requirements through multiple placement option alternatives

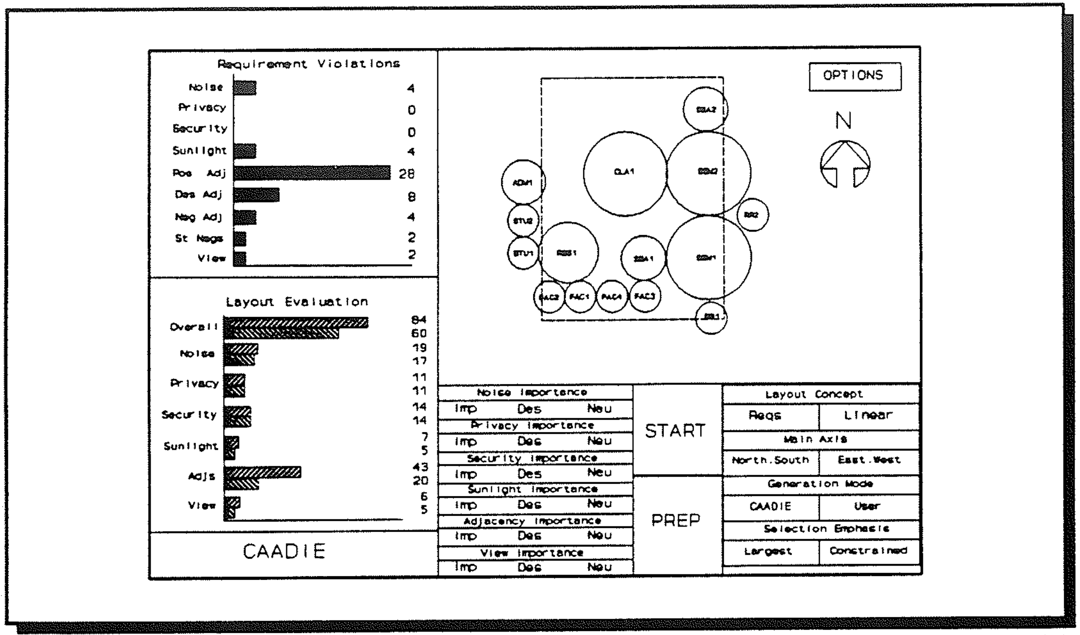


Figure B-7: The final CAADIE generated layout is displayed together with evaluation ratings

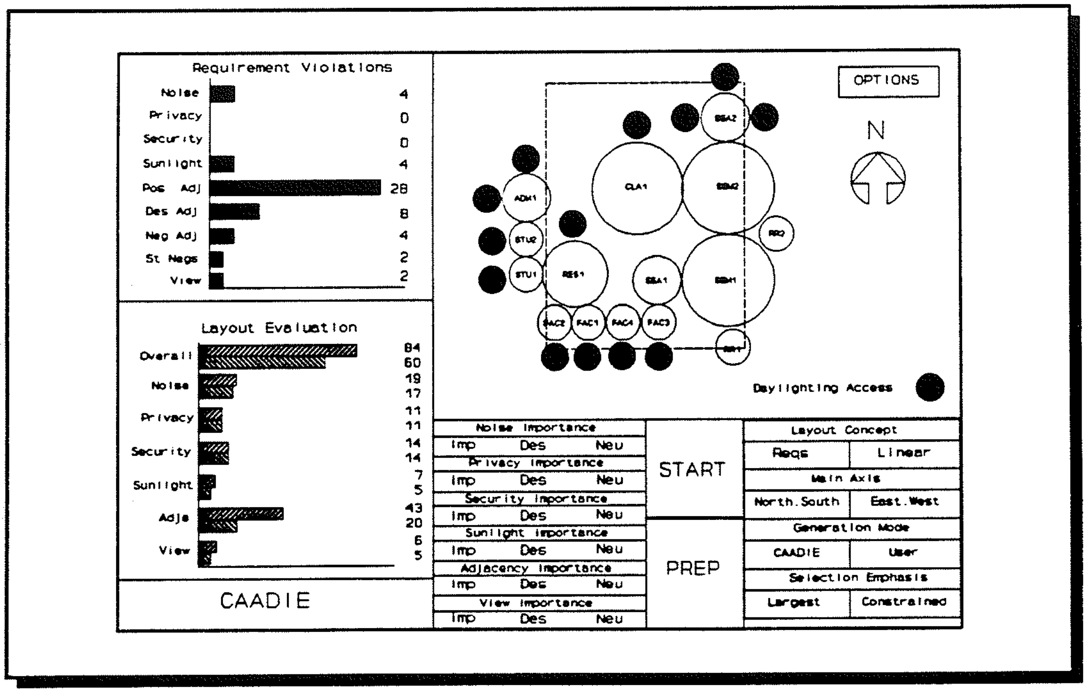


Figure B-8: A graphic requirement display portraying sunlight requirements

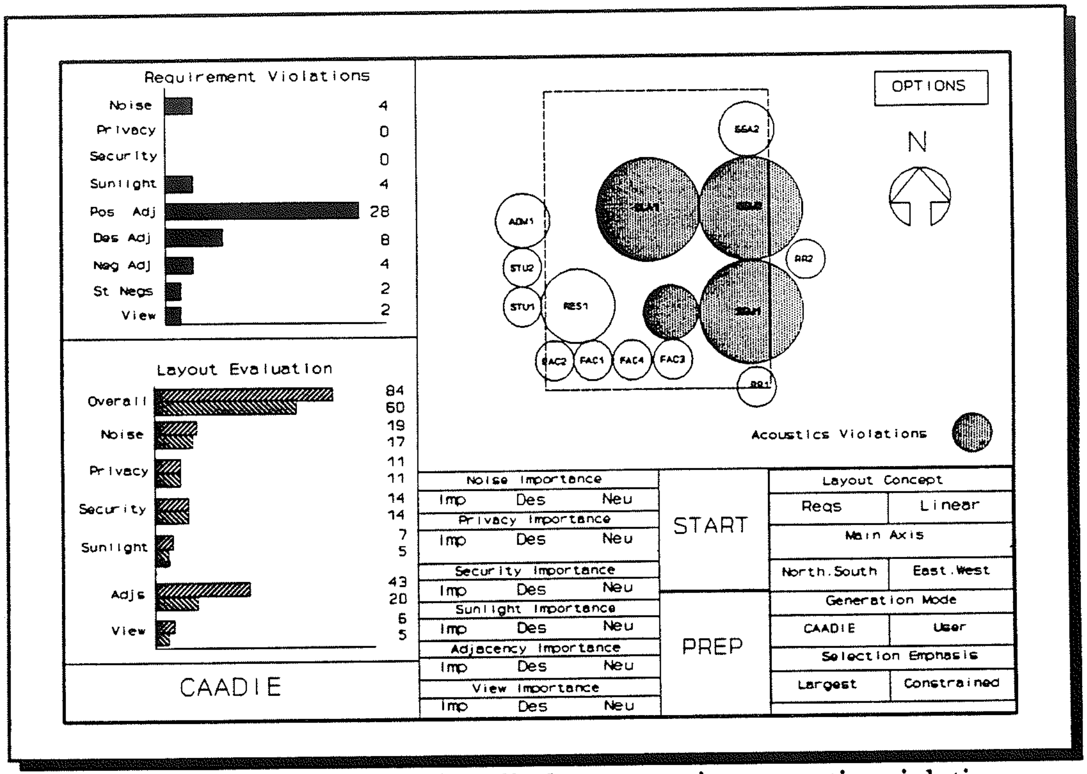


Figure B-9: An attribute violation display portraying acoustics violations

Direct Manipulation

The cooperative design capabilities available in the CAADIE user interface permit designers to manually manipulate layouts. Figures B-10 and B-11 illustrate the manipulation process for *administrative office 1*. In this example, the designer elects to place the administrative office closer to the four faculty offices. However, this placement results in an overlap condition with *restroom 1* and *seminar room 2*. CAADIE resolves this condition by altering the location of the affected spaces based on the minimum distance algorithm (see page 121). Upon completing the overlap resolution process, the system revises the evaluation charts to reflect the new attribute compliance ratings (figure B-12).

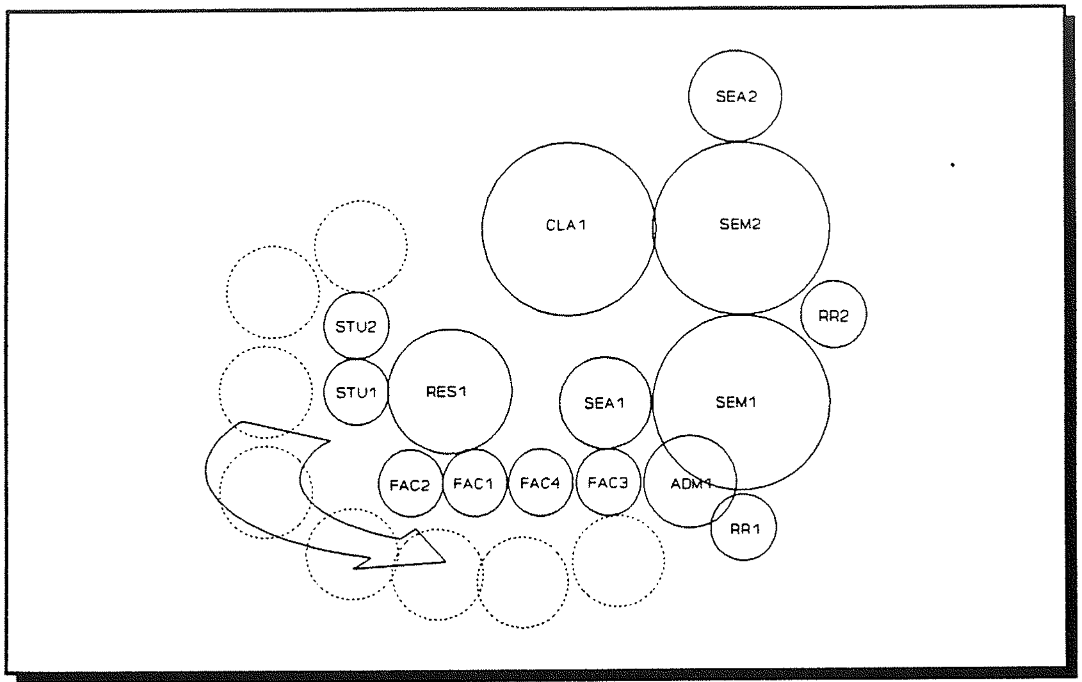


Figure B-10: Direct manipulation permits designers to manually alter spatial locations

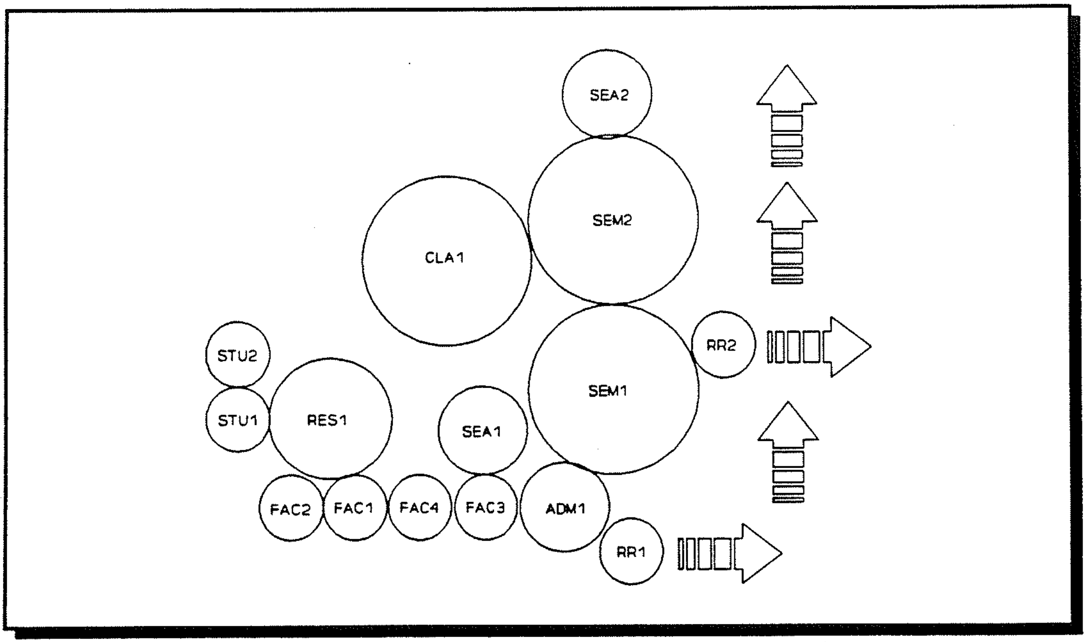


Figure B-11: CAADIE resolves overlap conditions by altering affected spaces based on the least movement algorithm

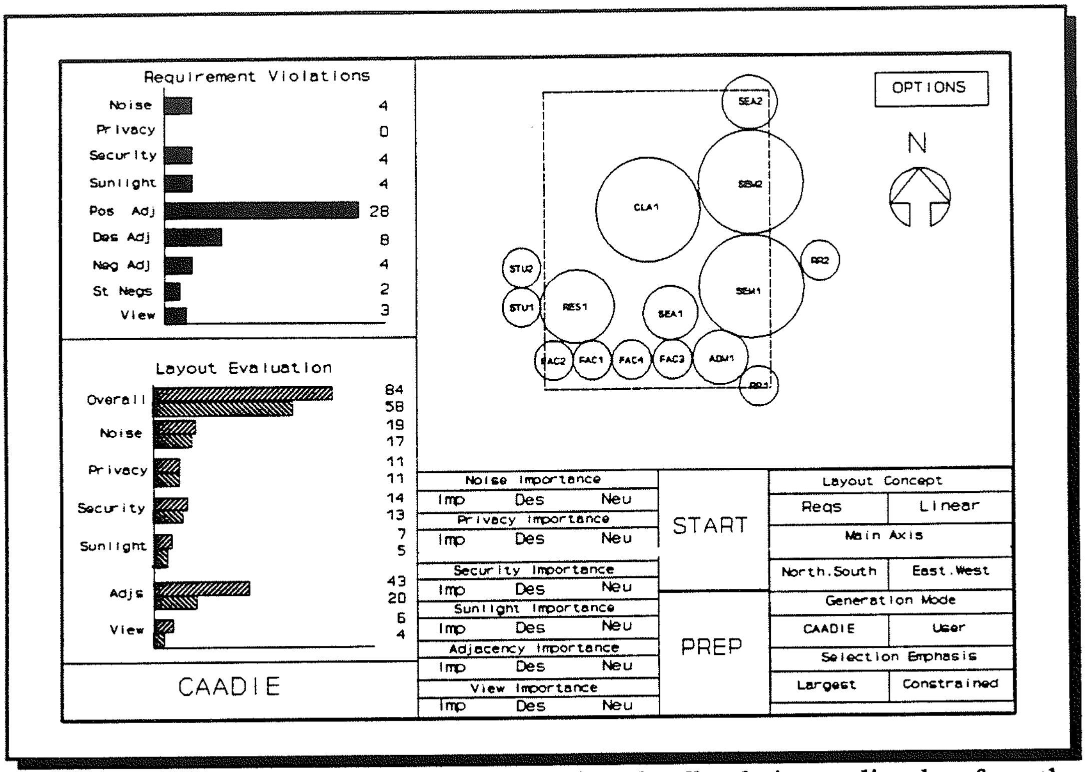


Figure B-12: CAADIE provides evaluation feedback immediately after the designer performs a direct manipulation

Attribute Manipulation

Attribute manipulation permits designers to explore the ramifications of altering design program requirements and general layout preferences. The current scenario illustrates the potential impact that a small number of layout requirement alterations can have on an overall configuration. Specifically, three positive adjacencies have been eliminated from the *seating area 2* requirements, and *adjacencies* has been added to the important attributes of *research lab 1*. The former alteration will provide greater flexibility to generate placement options for *seating area 2*, while the latter revision will restrict the placement of *research lab 1* by requiring additional evaluation procedures. Finally, the global preference

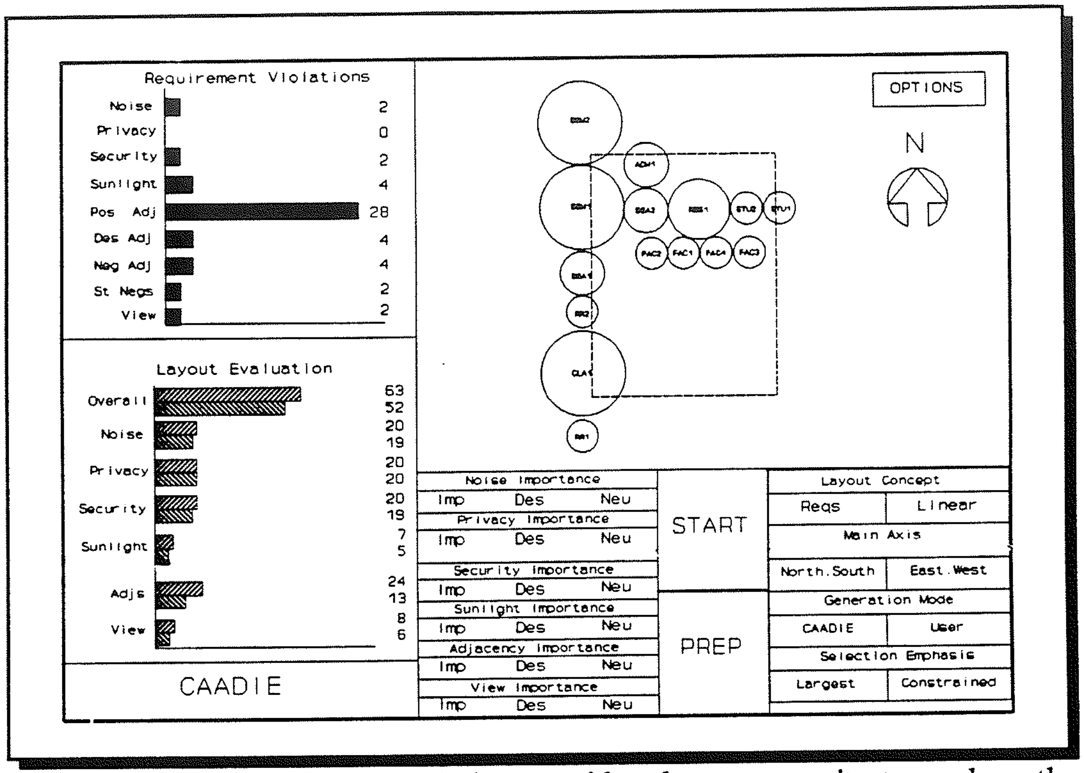


Figure B-13: Attribute manipulation provides the opportunity to explore the ramifications of altering layout requirements and preferences

factors have been reduced as follows to emphasize the space-specific attribute preferences:

Acoustics:	Neutral
Adjacencies:	Neutral
Privacy:	Desired
Security:	Desired
Sunlight:	Desired
View:	Important

Figure B-13 illustrates the results of regenerating the layout based on the revised layout requirements. The changes have resulted in altering the main concentration of spaces from the east side of the layout to the west side. Additionally, the attribute compliance percentage has increased by seven points (52/63 = 83% for the new layout versus 60/84 = 71% for the original layout).

However, based on the reduced preference factors, the actual layout rating has been reduced from 60% to 52%. This illustrates the trade-off between attribute compliance and overall layout ratings which occurs when designers alter the preference factors for individual attributes.

Scenario II - A Linear-Based Concept

The second scenario depicts the differences between selecting a requirements concept and a linear concept. The scenario encompasses the same fifteen spaces placed in the previous scenario. However, the focus attribute has been changed to security, and the attribute preference factors have been altered as follows:

Acoustics:	Important
Adjacencies:	Desired
Privacy:	Neutral
Security:	Important
Sunlight:	Desired
Views:	Important

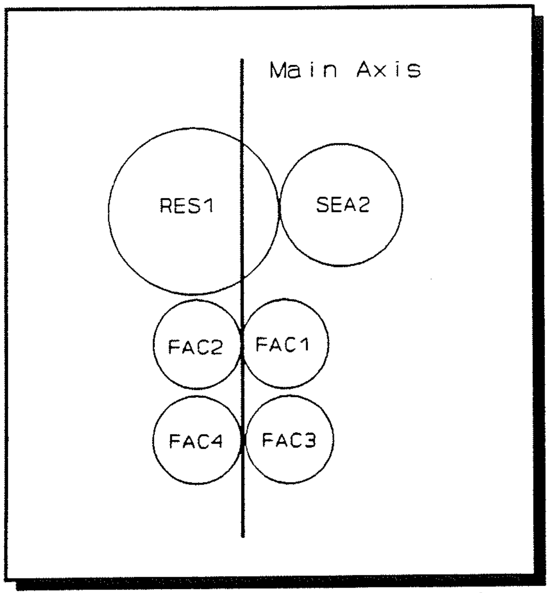


Figure B-14: The initial linear placements emphasize the main axis

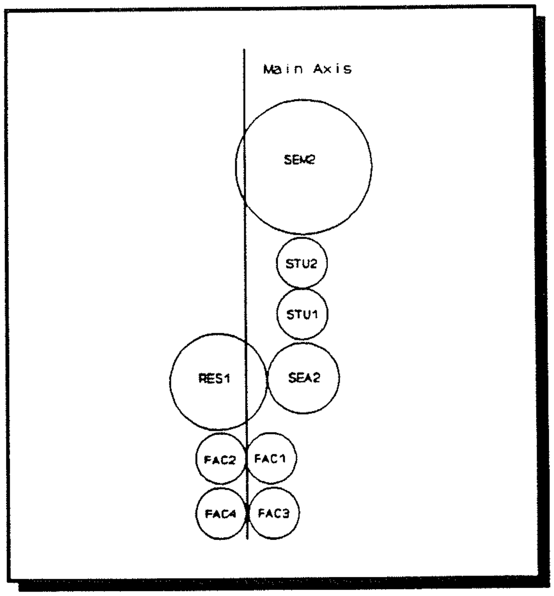


Figure B-15: The placement process continues to place spaces along the main axis to reinforce the concept

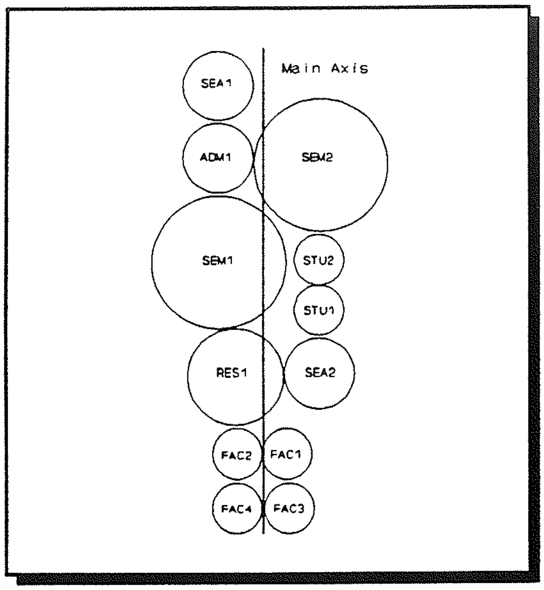


Figure B-16: Spaces are placed along the main axis until the area remaining does not accommodate the next space

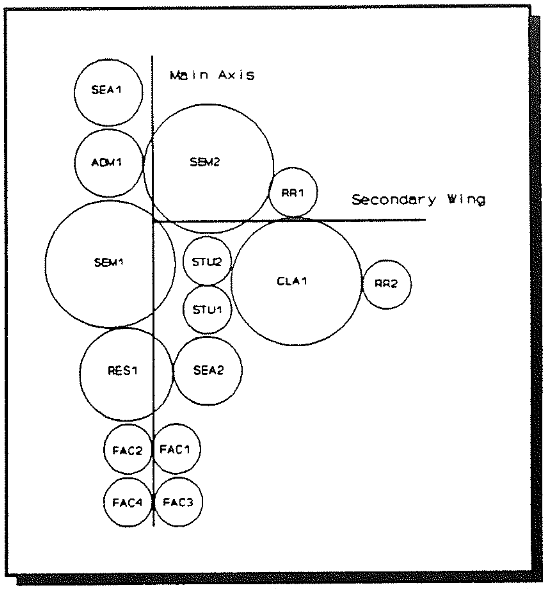


Figure B-17: A secondary axis is created to accommodate the overflow spaces

Figures B-14 through B-18 portray several layout generation stages as the linear concept is developed. The solution emphasizes a main north-south axis, with a secondary east-west wing to accommodate the overflow of spaces. In comparison to the requirements-based layout, the emphasis on form results in an increase in the number of non-adjacency attribute violations. This increase results from the reduced placement flexibility imposed by the form requirement. The reduction ensures that each placement option appropriately reinforces the linear concept. However, the limited number of placement locations reduces the possibility to comply with all layout requirements. Since adjacencies are used to guide the initial selection process, these requirements are emphasized by initial placement options. Based on the inability to create additional placement options, the remaining attributes will often be violated. Thus, in exchange for a linear emphasis, the resulting configuration contains a greater number of attribute violations.

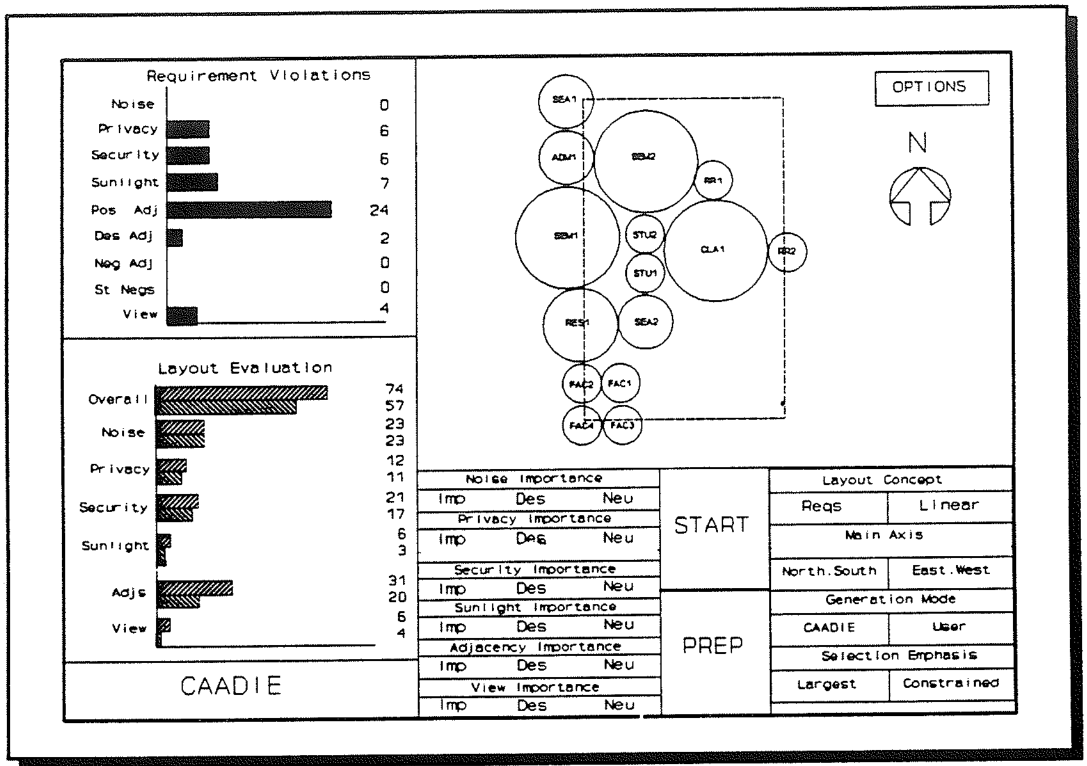


Figure B-18: The completed linear-based layout with accompanying evaluation feedback

Scenario III - A User Generated Layout

The final CAADIE scenario illustrates the sketch pad capabilities provided by the user interface. In this example, the designer generates a layout with the fifteen spaces used in the previous scenarios. As outlined in Chapter 6, the designer iterates through a selection-placement process to incrementally generate layout configurations (see page 120).

The system initializes the layout generation process by presenting the required spaces to the designer (figure B-19). The designer retains the flexibility to select and place these spaces in any order during the generation process. At each stage, CAADIE provides evaluation feedback based on the attribute compliance of each partial configuration (figures B-20 and B-21). This continuous evaluation permits attribute violations to be addressed as they occur. The designer can alter the partial configuration to resolve attribute violations, or elect to defer this alteration until the violations exist in a completed layout context.

Figure B-22 illustrates the user-generated configuration. The notable comparison between this configuration and the previous configuration resides in the reduced attribute compliance ratings. Cognitive limitations make it prohibitive for designers to concurrently address the multiple requirements impacting layout generation (Simon 1981). Furthermore, this limitation is amplified when multiple spaces are concerned. Thus, as the configuration expands beyond a few spaces, designers are forced to narrow their attribute compliance emphasis. As a result, it becomes increasingly difficult for designers to surpass the CAADIE compliance ratings as the number of spaces and number of layout attributes increase. It is this difficulty which highlights the utility of a layout generation assistant.

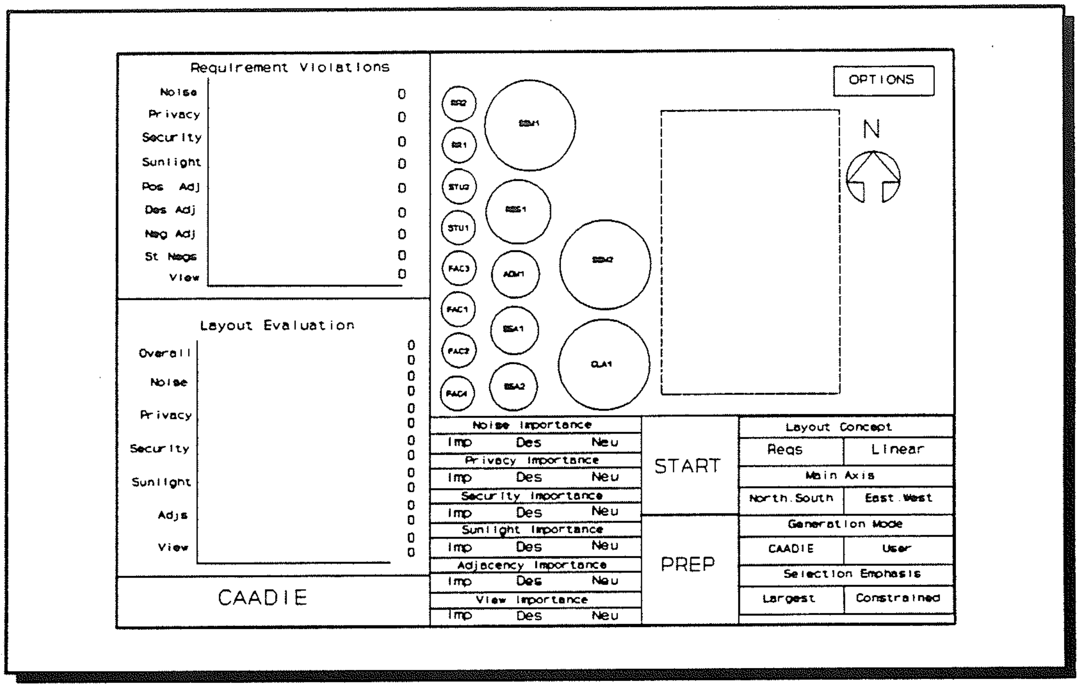


Figure B-19: CAADIE initially displays all required spaces to the designer

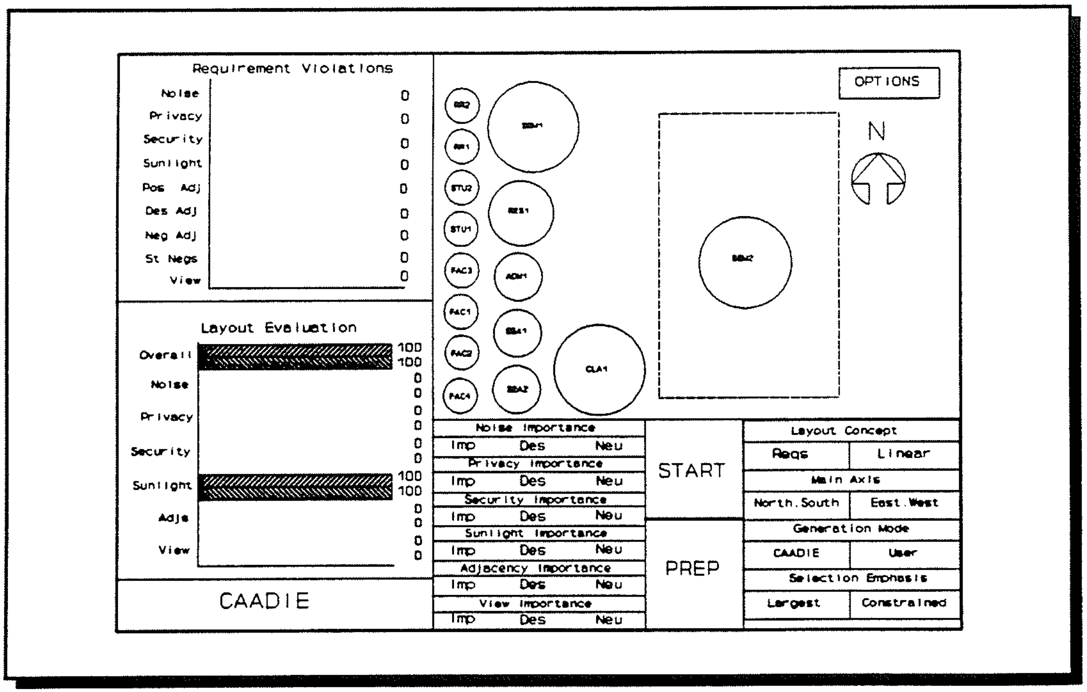


Figure B-20: The designer retains the flexibility to place the spaces in any order

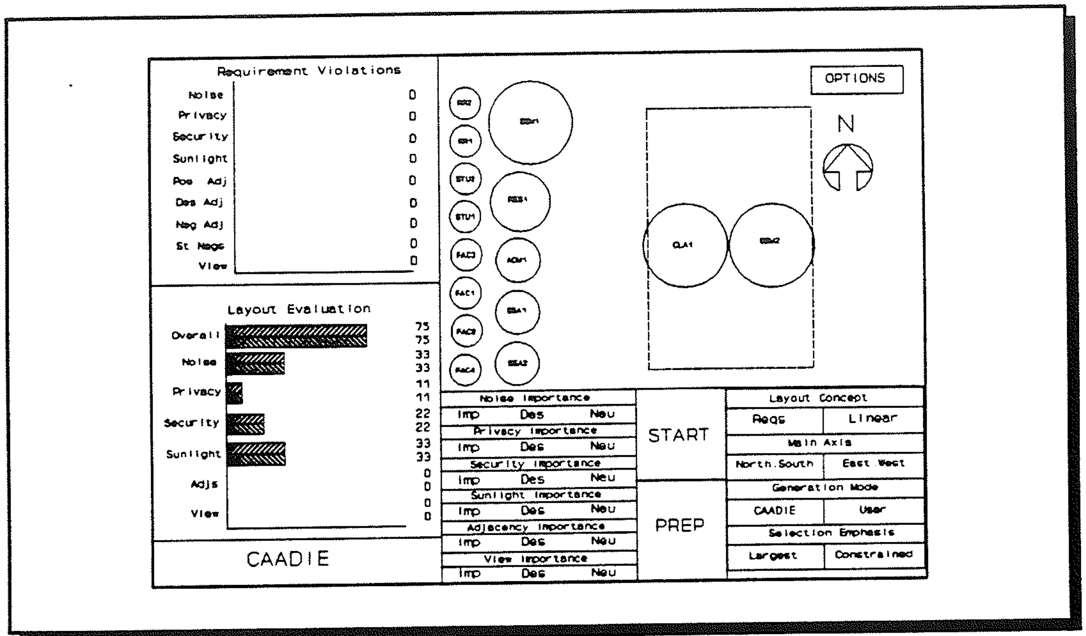


Figure B-21: CAADIE provides updated feedback as the designer places each space

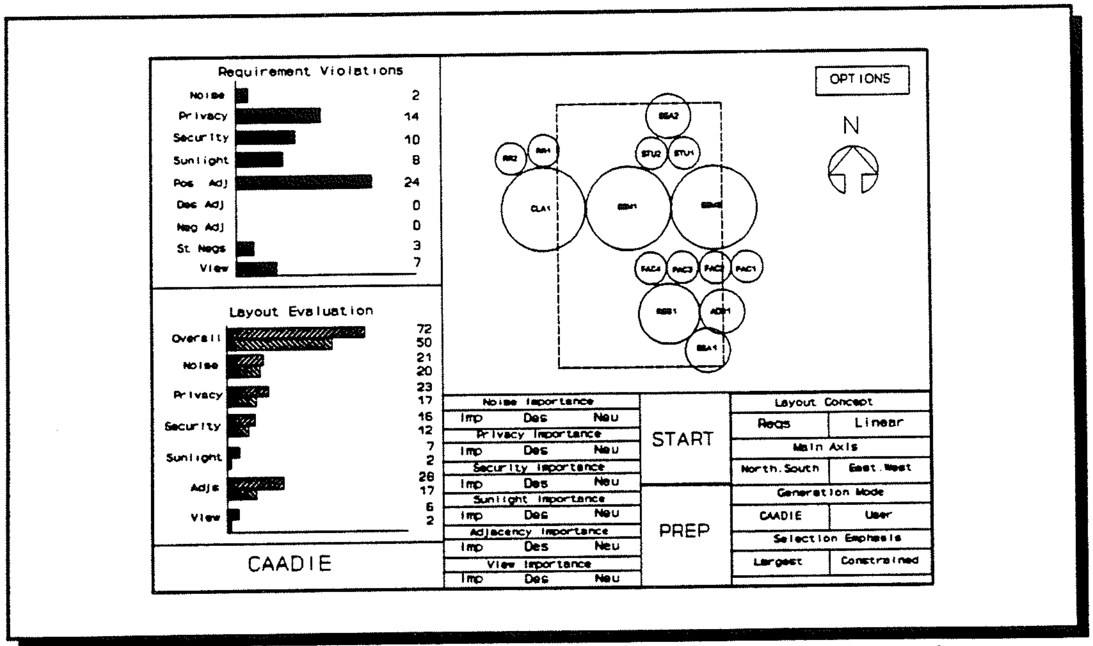


Figure B-22: The completed configuration with the evaluation ratings

APPENDIX C

CAADIE RULE SETS

This appendix details the CAADIE rule sets containing the knowledge selection heuristics and the spatial ordering heuristics. The heuristics included in these rule sets represent the core knowledge extracted from the knowledge acquisition sessions and the studies conducted by the author. It is intended, that these heuristics represent common design knowledge, and not, specific designer preferences. The designer expertise heuristics (i.e., the selection of spaces and conflict resolution) and design attribute heuristics (i.e., placement evaluation and validation) were previously introduced in Chapter 6. Thus, these heuristics are not repeated in this appendix.

Knowledge Source Selection Rules

Rule 1
IF
    The focus of an eligible knowledge source is equal to the focus of the previously executed knowledge source
THEN
    Increase the value of the eligible knowledge source

Rule 2
IF
    The subfocus of an eligible knowledge source is equal to the subfocus of the previously executed knowledge source
THEN
    Increase the value of the eligible knowledge source

Rule 3
IF
No eligible knowledge sources contain the same focus or subfocus as the previously executed knowledge source
THEN
Check the external or internal focus of the eligible knowledge sources

Rule 4
IF
The focus of the eligible knowledge source is external
THEN
Increase the value of the eligible knowledge source

Rule 5
IF
No eligible knowledge sources contain an external focus
THEN
Select the first eligible knowledge source with the highest score

Spatial Ordering Heuristics

Requirements-Based Concept

Rule 1
IF
The focus space contains a lighting requirement AND
No positive adjacencies of the next space, other than the focus space, have previously been placed
THEN
Attempt to place the next space adjacent to the focus space

Rule 2
IF
Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed
THEN
Attempt to place the next space adjacent to both the focus space and another positive adjacency

Rule 3
IF

Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
A positive adjacency contains the same lighting requirement as the next space AND
The next space cannot be placed adjacent to both the focus space and the positive adjacency with the same lighting requirement

THEN

Attempt to place the next space adjacent to the positive adjacency with the same lighting requirement

Rule 4
IF

Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
A positive adjacency contains the same lighting requirement as the next space AND
The next space cannot be placed adjacent to both the focus space and the positive adjacency with the same lighting requirement AND
The next space cannot be placed adjacent to the positive adjacency with the same lighting requirement

THEN

Attempt to place the next space adjacent to the focus space

Rule 5
IF

Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
No positive adjacency contains the same lighting requirement as the next space AND
The next space cannot be placed adjacent to both a positive adjacency and the focus space

THEN

Attempt to place the next space adjacent to the focus space

Rule 6
IF

Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
No positive adjacency contains the same lighting requirement as the next space AND
The next space cannot be placed adjacent to both a positive adjacency and the focus space AND
The next space cannot be placed adjacent to the focus space AND
A positive adjacency has been placed containing no lighting requirement

THEN

Attempt to place the next space adjacent to the positive adjacency with the same lighting requirement

Rule 7
IF

Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
No positive adjacency contains the same lighting requirement as the next space AND
The next space cannot be placed adjacent to both a positive adjacency and the focus space AND
The next space cannot be placed adjacent to the focus space AND
A positive adjacency has been placed containing no lighting requirement AND
The next space cannot be placed adjacent to the positive adjacency containing no lighting requirement AND
A previously placed space contains the same lighting requirement as the next space

THEN

Attempt to place the next space adjacent to the space containing the same lighting requirement as the next space

Rule 8
IF

Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
No positive adjacency contains the same lighting requirement as the next space AND

The next space cannot be placed adjacent to both a positive adjacency and the focus space AND
The next space cannot be placed adjacent to the focus space AND
A positive adjacency has been placed containing no lighting requirement AND
The next space cannot be placed adjacent to the positive adjacency containing no lighting requirement AND
No space has been placed which contains the same lighting requirement as the next space AND
A space has been placed which contains no lighting requirement

THEN

Attempt to place the next space adjacent to the space containing no lighting requirement

Rule 9

IF

Both the focus space and the next space contain lighting requirements AND
No positive adjacencies of the next space, other than the focus space, have previously been placed

THEN

Attempt to place the next space adjacent to the focus space

Rule 10

IF

Both the focus space and the next space contain lighting requirements AND
No positive adjacencies of the next space, other than the focus space, have previously been placed AND
The next space cannot be placed adjacent to the focus space AND
A previously placed space contains the same lighting requirement as the next space

THEN

Attempt to place the next space adjacent to the space containing the same lighting requirement

Rule 11

IF

Both the focus space and the next space contain lighting requirements AND
No positive adjacencies of the next space, other than the focus space, have previously been placed AND
The next space cannot be placed adjacent to the focus space AND
No previously placed spaces contain the same lighting requirement as the next space AND
A previously placed space contains no lighting requirement

THEN

Attempt to place the next space adjacent to the space containing no lighting requirement

Rule 12

IF

Both the focus space and the next space contain lighting requirements AND
No positive adjacencies of the next space, other than the focus space, have previously been placed AND
The next space cannot be placed adjacent to the focus space AND
A previously placed space contains no lighting requirement

THEN

Attempt to place the next space adjacent to the space containing no lighting requirement

Rule 13

IF

Both the focus space and the next space contain lighting requirements AND
No positive adjacencies of the next space, other than the focus space, have previously been placed AND
The next space cannot be placed adjacent to the focus space AND
No previously placed spaces contain the same lighting requirement as the next space AND
A previously placed space contains no lighting requirement AND
The next space cannot be placed adjacent to the space containing no lighting requirement

THEN

Attempt to place the next space adjacent to both the focus space and a space adjacent to the focus space

Rule 14

IF

Both the focus space and the next space contain lighting requirements AND
No positive adjacencies of the next space, other than the focus space, have previously been placed AND
The next space cannot be placed adjacent to the focus space AND
No previously placed spaces contain the same lighting requirement as the next space AND
All previously placed spaces contain lighting requirements

THEN

Attempt to place the next space adjacent to both the focus space and a space adjacent to the focus space

Rule 15
IF
    Both the focus space and the next space contain lighting requirements AND
    The focus space is the only previously placed positive adjacency of the next space AND
    The focus space contains the same lighting requirement as the next space
THEN
    Attempt to place the next space adjacent to the focus space

Rule 16
IF
    Both the focus space and the next space contain lighting requirements AND
    A positive adjacency of the next space, other than the focus space, has previously been placed AND
    No previously placed positive adjacencies contain the same lighting requirement as the next space
THEN
    Attempt to place the next space adjacent to both the focus space and a positive adjacency

Rule 17
IF
    The next space contains no lighting requirement OR
    The focus space contains no lighting requirement AND
    The focus space is the only previously placed positive adjacency of the next space
THEN
    Attempt to place the next space adjacent to the focus space

Rule 18
IF
    Both the focus space and the next space contain lighting requirements AND
    A positive adjacency of the next space, other than the focus space, has previously been placed AND
    The previously placed positive adjacency contains no lighting requirement
THEN
    Attempt to place the next space adjacent to the focus space and the positive adjacency containing no lighting requirement

Rule 19
IF
Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
The previously placed positive adjacency contains no lighting requirement AND
The next space cannot be placed adjacent to the focus space and the positive adjacency containing no lighting requirement
THEN
Attempt to place the next space adjacent to the positive adjacency containing no lighting requirement

Rule 20
IF
Both the focus space and the next space contain lighting requirements AND
No positive adjacencies of the next space, other than the focus space, have previously been placed AND
The next space cannot be placed adjacent to the focus space
THEN
Attempt to place the next space adjacent to a space which is adjacent to the focus space

Rule 21
IF
Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
A positive adjacency contains the same lighting requirement as the next space AND
The next space cannot be placed adjacent to both the focus space and the positive adjacency containing the same lighting requirement as the next space
THEN
Attempt to place the next space adjacent to the positive adjacency containing the same lighting requirement as the next space

Rule 22
IF
Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND

A positive adjacency contains the same lighting requirement as the next space
AND
The next space cannot be placed adjacent to both the focus space and the positive adjacency containing the same lighting requirement as the next space
AND
The next space cannot be placed adjacent to the positive adjacency containing the same lighting requirement as the next space

THEN
Attempt to place the next space adjacent to the focus space

Rule 23
IF
Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
A positive adjacency contains the same lighting requirement as the next space

THEN
Attempt to place the next space adjacent to both the focus space and the positive adjacency containing the same lighting requirement as the next space

Rule 24
IF
The next space contains a lighting requirement AND
The focus space does not contain a lighting requirement AND
No positive adjacencies of the next space, other than the focus space, have previously been placed

THEN
Attempt to place the next space adjacent to the focus space

Rule 25
IF
The next space contains a lighting requirement AND
The focus space does not contain a lighting requirement AND
No positive adjacencies of the next space, other than the focus space, have previously been placed AND
The next space cannot be placed adjacent to the focus space

THEN
Attempt to place the next space adjacent to a space which is adjacent to the focus space

Rule 26

IF

The next space contains a lighting requirement AND

The focus space does not contain a lighting requirement AND

A positive adjacency of the next space, other than the focus space, has previously been placed AND

No positive adjacency contains the same lighting requirement as the next space

THEN

Attempt to place the next space adjacent to a space which contains the lighting requirement as the next space

Rule 27

IF

The next space contains a lighting requirement AND

The focus space does not contain a lighting requirement AND

A positive adjacency of the next space, other than the focus space, has previously been placed AND

No positive adjacency contains the same lighting requirement as the next space AND

The next space cannot be placed adjacent to a space which contains the same lighting requirement as the next space AND

The next space cannot be placed adjacent to the focus space

THEN

Attempt to place the next space adjacent to a space which is adjacent to the focus space

Rule 28

IF

The focus space contains a lighting requirement AND

The next space does not contain a lighting requirement AND

A positive adjacency of the next space, other than the focus space, has previously been placed AND

A previously placed positive adjacency contains a lighting requirement

THEN

Attempt to place the next space adjacent to the focus space and a positive adjacency containing a lighting requirement

Rule 29

IF

The focus space contains a lighting requirement AND

The next space does not contain a lighting requirement AND

A positive adjacency of the next space, other than the focus space, has previously been placed AND
A previously placed positive adjacency contains a lighting requirement AND
The next space cannot be placed adjacent to the focus space and a positive adjacency containing a lighting requirement

THEN
Attempt to place the next space adjacent to the focus space

Rule 30
IF
The focus space contains a lighting requirement AND
The next space does not contain a lighting requirement AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
A previously placed positive adjacency contains a lighting requirement AND
The next space cannot be placed adjacent to the focus space and a positive adjacency containing a lighting requirement AND
The next space cannot be placed adjacent to the focus space

THEN
Attempt to place the next space adjacent to the positive adjacency containing the lighting requirement

Rule 31
IF
The focus space contains a lighting requirement AND
The next space does not contain a lighting requirement AND
A positive adjacency of the next space, other than the focus space, has previously been placed

THEN
Attempt to place the next space adjacent to the focus space and a positive adjacency

Rule 32
IF
The focus space contains a lighting requirement AND
The next space does not contain a lighting requirement AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
The next space cannot be placed adjacent to both the focus space and a positive adjacency

THEN
Attempt to place the next space adjacent to the focus space

Rule 33
IF

Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
No previously placed positive adjacencies contain the same lighting requirement as the next space AND
The next space cannot be placed adjacent to both the focus space and a positive adjacency AND
The next space cannot be placed adjacent to the focus space AND
All previously placed spaces contain a lighting requirement AND
No previously placed spaces contain the same lighting requirement as the next space AND
The next space cannot be placed adjacent to a positive adjacency

THEN

Attempt to place the next space adjacent to a previously placed space

Rule 34
IF

Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
No previously placed positive adjacencies contain the same lighting requirement as the next space AND
The next space cannot be placed adjacent to both the focus space and a positive adjacency AND
The next space cannot be placed adjacent to the focus space AND
The next space cannot be placed adjacent to a space containing no lighting requirement

THEN

Attempt to place the next space adjacent to a previously placed space

Rule 35
IF

Both the focus space and the next space contain lighting requirements AND
A positive adjacency of the next space, other than the focus space, has previously been placed AND
No previously placed positive adjacencies contain the same lighting requirement as the next space AND
The next space cannot be placed adjacent to both the focus space and a positive adjacency AND
A previously placed space contains no lighting requirement

THEN
    Attempt to place the next space adjacent to both the focus space and a space containing no lighting requirement

Linear-Based Concept

Rule 1
IF
    Space is available on the main axis AND
    The side across from the focus space on the main axis is available
THEN
    Attempt to place the next space adjacent to the focus space, across the main axis

Rule 2
IF
    Space is available on the main axis AND
    The side across from the focus space on the main axis is not available AND
    An adjacent side of the focus space is available on the same side of the main axis
THEN
    Attempt to place the next space adjacent to the focus space on the same side of the main axis

Rule 3
IF
    Space is available on the main axis AND
    The side across from the focus space on the main axis is not available AND
    The adjacent side of the focus space on the same side of the main axis is not available AND
    The side across the main axis from a space adjacent to the focus space is available
THEN
    Attempt to place the next space adjacent to a space which is adjacent to the focus space on the side across the main axis

Rule 4
IF
    Space is available on the main axis AND
    The side across from the focus space on the main axis is not available AND

The adjacent side of the focus space on the same side of the main axis is not available AND
The side across the main axis from a space adjacent to the focus space is not available
THEN
Attempt to place the next space adjacent to a remaining space on the main axis

Rule 5
IF
Space is available on the main axis AND
No placement options on the main axis comply with all of the stated design program requirements
THEN
Retry the previous rules without checking attribute compliance for the next space

Rule 6
IF
The space available on the main axis is not adequate for the next space AND
Space is available adjoining the focus space
THEN
Attempt to start a new wing adjacent to the focus space

Rule 7
IF
The space available on the main axis is not adequate for the next space AND
Space is not available adjoining the focus space AND
Space is available adjoining a space which is adjacent to the focus space
THEN
Attempt to start a new wing adjacent to the space which is adjacent to the focus space

Rule 8
IF
The space available on the main axis is not adequate for the next space AND
Space is not available adjoining the focus space AND
Space is not available adjoining a space which is adjacent to the focus space
THEN
Start a new wing adjacent to any available space on the main axis

Rule 9
IF
If the current axis is not the main axis AND
The side across from the focus space on the current axis is available
THEN
Attempt to place the next space adjacent to the focus space, across the current axis

Rule 10
IF
If the current axis is not the main axis AND
The side across from the focus space on the current axis is not available AND
An adjacent side of the focus space is available on the same side of the current axis
THEN
Attempt to place the next space adjacent to the focus space on the same side of the current axis

Rule 11
IF
If the current axis is not the main axis AND
The side across from the focus space on the current axis is not available AND
An adjacent side of the focus space is not available on the same side of the current axis AND
The side across the current axis from a space adjacent to the focus space is available
THEN
Attempt to place the next space adjacent to a space which is adjacent to the focus space on the side across the current axis

Rule 12
IF
If the current axis is not the main axis AND
The side across from the focus space on the current axis is not available AND
An adjacent side of the focus space is not available on the same side of the current axis AND
The side across the current axis from a space adjacent to the focus space is not available

THEN
Attempt to place the next space adjacent to a remaining space on the current axis

Rule 13
IF
    If the current axis is not the main axis AND
    No positions are available on the current axis OR
    No room is remaining on the current axis
THEN
    Start a new wing from the main axis

BIBLIOGRAPHY

Akin, Omer. "How do Architects Design?." In Artificial Intelligence and Pattern Recognition in Computer Aided Design, ed. J.C. Latombe, 65-97. Amsterdam: North-Holland Publishing Company, 1978.

Akin, Omer. "Exploration of the Design Process." Design Methods and Theories 13, Number 3/4 (1979): 115-118.

Akin, Omer. "Expertise of the Architect." In Expert Systems for Engineering Design, ed. Michael D. Rychener, 173-195. Boston: Academic Press, Inc., 1988.

Akin, Omer, B. Dave, and S. Pithavadian. "Heuristic generation of layouts (HeGeL): Based on a paradigm for problem structuring." In Artificial Intelligence in Engineering Design, ed. John S. Gero, 413-444. Southampton: Computational Mechanics, 1988.

Akiner, V. Tuncer. "Topology-1: A Knowledge-Based System for Reasoning About Objects and Spaces." Design Studies 7, Number 2 (1986): 94-104.

Alexander, Christopher and Barry Poyner. "The Atoms of Environmental Structure." In Emerging Methods in Environmental Design and Planning, ed. Gary T. Moore, 308-321. Cambridge: MIT Press, 1970.

Armour, Gordon C. and Elwood S. Buffa. "A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities." Journal of the Institute of Management Sciences 9, Number 2 (1963): 294-309.

Baybars, I. and Charles M. Eastman. "Enumerating Architectural Arrangements by Generating Their Underlying Graphs." Environment and Planning B 7, Number 3 (1980): 289-310.

Bloch, C.J. and R. Krishnamurti. "The Counting of Rectangular Dissections." Environment and Planning B 5, Number 2 (1978): 207-214.

Bond, Laura S., Kenneth H. Crawford, Beth A. Symonds, James R. Anderson, and Sue Weideman. A Case Study of a Prototype Computer-Aided Architectural Design System. Champagne: US Army Corps of Engineers Construction Engineering Research Laboratory, 1988. USA-CERL, P-88/17.

Buchanan, Bruce G., David Barstow, Robert Bechtal, James Bennett, William Clancey, Casimir Kulikowski, Tom Mitchell, and Donald A. Waterman. "Constructing an Expert System." In Building Expert Systems, ed. Frederick Hayes-Roth, Donald A. Waterman, and Douglas B. Lenat, 127-167. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., 1983.

Castaldi, Basil. Educational Facilities: Planning, Modernization, and Management, Boston: Allyn and Bacon, Inc., 1987.

Chase, Laura, Architect, Carrasco and Associates. Interview by author, September 1990, Carrasco and Associates, Palo Alto.

Chase, S.C. "Shapes and Shape Grammars: from Mathematical Model to Computer Implementation." Environment and Planning B 16, Number 3 (1989): 215-242.

Ching, Frank. Architecture, Form, Space, and Order, New York: Van Nostrand Reinhold, 1979.

Cinar, Unver. "Facilities Planning: A Systems Analysis and Space Allocation Approach." In Spatial Synthesis in Computer-Aided Building Design, ed. Charles M. Eastman, 19-39. New York: Halsted Press, 1975.

Coyne, Richard D., and John S. Gero. "Design Knowledge and Sequential Plans." Environment and Planning B 12, Number 4 (1985): 401-418.

Coyne, Richard D., and John S. Gero. "Design Knowledge and Context." Environment and Planning B 12, Number 4 (1985): 419-442.

Coyne, Richard D., Michael A. Rosenman, Anthony D. Radford, M. Balachandran, and John S. Gero. Knowledge-Based Design Systems. Reading, Massachusetts: Addison-Wesley Publishing Company, 1990.

Coyne, Robert F., and Ulrich Flemming. "Planning in Design Synthesis: Abstraction-Based LOOS." In Proceedings of AIENG 90 held in Boston, MA, July 1990, 1990.

de la Croix, Horst and Richard G. Tansey. Art Through the Ages. New York: Harcourt Brace Jovanovich, Inc., 1980.

Duce, D.A., and G.A. Ringland. Approaches to Knowledge Representation: An Introduction. Hertshire, England: Research Studies Press Limited, 1988.

Dudnik, Elliot E. and Robert Krawczyk. "An Evaluation of Space Planning Methodologies." In Proceedings of the Fourth International EDRA Conference Held at Virginia Polytechnic University 15-18 April 1973, edited by Wolfgang F E Preiser, 414-427. Stroudsburg, PA: Dowden, Hutchinson, and Ross, Inc., 1973.

Duffy, Francis. "Office Design and Organizations: 1. Theoretical Basis." Environment and Planning B 1, Number 2 (1974): 105-118.

Duffy, Francis and John Torrey. "A Progress Report on the Pattern Language." In Emerging Methods in Environmental Design and Planning, ed. Gary T. Moore, 261-277. Cambridge: MIT Press, 1970.

Dym, C.L., R.P. Henchey, E.A. Delis, and S. Gonick. "A Knowledge-Based System for Automated Architectural Code Checking." Computer Aided Design 20, Number 3 (1988): 137-145.

Earl, C.F. "Rectangular Shapes." Environment and Planning B 7, Number 3 (1980): 311-342.

Eastman, Charles M. "On the Analysis of Intuitive Design Processes." In Emerging Methods in Environmental Design and Planning, ed. Gary T. Moore, 21-37. Cambridge: MIT Press, 1970.

Eastman, Charles M. "Problem Solving Strategies in Design." In Proceedings of the First Annual EDRA Conference Held at the University of North Carolina June 1969, edited by Henry Sanoff and Sidney Cohn, 138-153. Stroudsburg, PA: Dowden, Hutchinson, and Ross, Inc., 1970.

Eastman, Charles M. "Requirements for Men-Machine Collaboration in Design." In Proceedings of the Fourth International EDRA Conference Held at Virginia Polytechnic University 15-18 April 1973, edited by Wolfgang F E Preiser, 396-413. Stroudsburg, PA: Dowden, Hutchinson, and Ross, Inc., 1973.

Eastman, Charles M. "The Scope of Computer-Aided Building Design." In Spatial Synthesis in Computer-Aided Building Design, ed. Charles M. Eastman, 1-18. New York: Halsted Press, 1975.

Erman, Lee D., Frederick Hayes-Roth, Victor R. Lesser, and D. Raj Reddy. "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty." Computing Surveys 12, Number 2 (1980): 213-252.

Fawcett, William H. "Design Knowledge in Architectural CAD." Computer Aided Design 18, Number 2 (1986): 83-90.

Fenves, Steven J., Ulrich Flemming, Chris T. Hendrickson, Mary Lou Maher, and Gerhard Schmitt. "A Prototype Environment for Integrated Design and Construction Planning of Buildings." In Proceedings of the first Symposium of the Center for Integrated Facilities Engineering Held at Stanford University 26-29 March 1989, by the Center for Integrated Facilities Engineering. Stanford: Center for Integrated Facilities Engineering, 1989.

Fikes, R., and T. Kehler. "The Role of Frame-Based Representation in Reasoning." Communications of the ACM 28, Number 9 (1985): 904-920.

Flemming, Ulrich. "Wall Representations of Rectangular Dissections and Their Use in Automated Space Allocation." Environment and Planning B 5, Number 3 (1978): 215-232.

Flemming, Ulrich. "On the Representation and Generation of Loosely Packed Arrangements of Rectangles." Environment and Planning B 13, Number 2 (1986): 189-206.

Flemming, Ulrich. "Rule-Based Systems in Computer-Aided Architectural Design." In Expert Systems for Engineering Design, ed. Michael D. Rychener, 93-111. Boston: Academic Press, Inc., 1988.

Flemming, Ulrich. "More on the Representation and Generation of Loosely-Packed Arrangements of Rectangles." Environment and Planning B 16, Number 3 (1989): 249-366.

Flemming, Ulrich, Robert F. Coyne, Timothy Glavin, Hung Hsi, and Michael D. Rychener. A Generative Expert System for the Design of Building Layouts. Pittsburgh: Engineering Design Research Center, Carnegie-Mellon University, 1988.

Galle, P. "Abstraction as a Tool of Automated Floor-Plan Design." Environment and Planning B 13, Number 1 (1986): 21-41.

Garcia, Ana Cristina Bicharra, and H. Craig Howard. "Building a Model for Augmented Design Documentation." To appear in Proceedings of the First International Conference on AI in Design, to be held in Edinburgh, Scotland, June 1991, to be edited by J.S. Gero, 1991.

Gero, John S. "Design Prototypes: A Knowledge Representation Schema for Design." The AI Magazine 11, Number 4 (1990): 26-36.

Gero, John S. and Richard D. Coyne. "Logic Programming as a Means of Representing Semantics in Design Languages." Environment and Planning B 12, Number 3 (1985a): 351-369.

Gero, John S. and Richard D. Coyne. "Design Knowledge and Sequential Plans." Environment and Planning B 12, Number 3 (1985b): 401-418.

Gero, John, Mary Lou Maher, and Weiguang Zhang. Chunking Structural Design Knowledge as Prototypes. Sydney: The Architectural Computing Unit, Department of Architectural Science, University of Sydney, 1988. Working Paper.

Gilleard, J. "LAYOUT - A Hierarchical Computer Model for the Production of Architectural Floor Plans." Environment and Planning B 5, Number 2 (1978): 233-242.

Gips, J. and G. Stiny. "Production Systems and Grammars: A Uniform Characterization." Environment and Planning B 7, Number 4 (1980): 399-408.

Grant, Donald P. "Space Planning Methods: Part One - A Review of Reviews from the Nineteen-Seventies." Design Methods and Theories 17, Number 1 (1983a): 4-33.

Grant, Donald P. "A General Morphology of Systematic Space Planning Approaches" Design Methods and Theories 17, Number 2 (1983b): 57-84.

Grant, Donald P. "Space Planning Methods: Part Three - A Review of the Literature on Overlay Models for Space Planning." Design Methods and Theories 17, Number 3 (1983c): 118-142.

Grason, John. "Fundamental Description of a Floor Plan Design Program." In Proceedings of the First Annual EDRA Conference Held at the University of North Carolina June 1969, edited by Henry Sanoff and Sidney Cohn, 175-180. Stroudsburg, PA: Dowden, Hutchinson, and Ross, Inc., 1970a.

Grason, John. "A Dual Linear Graph Representation for Space-Filling Location Problems of the Floor Plan Type." In Emerging Methods in Environmental Design and Planning, ed. Gary T. Moore, 21-37. Cambridge: MIT Press, 1970b.

Gross, Mark D., Stephen M. Ervin, James A. Anderson, and Aaron Fleisher. "Constraints: Knowledge Representation in Design." Design Studies 9, Number 3 (1988): 133-143.

Harrigan, John E., and Janet R. Harrigan. Human Factors Program for Architects, Interior Designers and Clients. San Luis Obispo, CA: Blake Printery, 1979.

Hart, Anna. "Knowledge Elicitation: Issues and Methods." Computer Aided Design 17, Number 9 (1985): 455-462.

Hashimshony, R. and J. Roth. "ALG: A Model for Generating Alternative Layout Graphs Under Architectural Constraints." Computer Aided Design 18, Number 8 (1986): 431-436.

Hawkes, Dean. "Types, Norms and Habit in Environmental Design." In The Architecture of Form, ed. Lionel March, 465-481. Cambridge: Cambridge University Press, 1976.

Hayes-Roth, Barbara. BB1: An Architecture for Blackboard Systems that Control, Explain, and Learn About Their Own Behavior. Knowledge Systems Laboratory, Computer Science Department, Stanford University, 1984. HPP 84-16.

Hayes-Roth, Barbara. "A Blackboard Architecture for Control." Artificial Intelligence 26 (1985): 251-321.

Hayes-Roth, Barbara. Making Intelligent Systems Adaptive. Knowledge Systems Laboratory, Computer Science Department, Stanford University, (July 1988).

Heath, Thomas F. "The Algorithmic Nature of the Design Process." In Emerging Methods in Environmental Design and Planning, ed. Gary T. Moore, 57-69. Cambridge: MIT Press, 1970.

Henrion, Max. "Automatic Space-Planning: A Postmortem?" In Artificial Intelligence and Pattern Recognition in Computer Aided Design, ed. J.C. Latombe, 175-191. Amsterdam: North-Holland Publishing Company, 1978.

Hyde, Richard. "Design Procedures in Architectural Design: Applications in CAAD." Design Studies 10, Number 4 (1989): 239-245.

International Council of Educational Facility Planners. Space Planning Guidelines. July 1985.

Ito, Kenji, Kincho H. Law, and Raymond E. Levitt. PMAPM: An Object-Oriented Project Model for A/E/C Process with Multiple Views. Center for Integrated Facility Engineering, Stanford University, 1990. Technical Report 34.

Jackson, Peter. Introduction to Expert Systems. Reading, MA.: Addison-Wesley Publishers Limited, 1986.

Johnson, M. Vaughn, Jr., and Barbara Hayes-Roth. Integrating Diverse Reasoning Methods in the BB1 Blackboard Control Architecture. Knowledge Systems Laboratory, Computer Science Department, Stanford University, 1986. KSL 86-76.

Jones, J. Christopher. "The State of the Art in Design Methods." In Emerging Methods in Environmental Design and Planning, ed. Gary T. Moore, 3-8. Cambridge: MIT Press, 1970.

Jones, J. Christopher. "Designing Designing." Design Studies 1, Number 1 (1979): 31-35.

Knight, T.W. "Color Grammars: Designing with Lines and Colors." Environment and Planning B 16, Number 4 (1989): 417-449.

Koopmans, T.C. and J. Beckmann. "Assignment Problems and the Location of Economic Activities." Econometrica 25, Number 1 (1957): 53-76.

Krauss, Richard I., John R. Myer, Scott Danielson, and Roger Lewis. "Design: A Case History." In Emerging Methods in Environmental Design and Planning, ed. Gary T. Moore, 11-20. Cambridge, MA: MIT Press, 1970.

Lee, R.C. and J.M. Moore. "CORELAP - Computerized Relationship Layout Planning." Journal of Industrial Engineering 8, Number 3 (1967): 195-200.

Lenat, D.B., and R.V. Guha. Building Large Knowledge-Based Systems. Reading, MA: Addison-Wesley Publishing Company, Inc., 1990.

Levitt, Raymond E., Iris Tommelein, Barbara Hayes-Roth, and Tony Confrey. SightPlan: An Expert System that Models and Augments Human Decision-Making for Designing Construction Site Layouts. Center for Integrated Facility Engineering, Stanford University, 1990. Technical Report 20.

Lew, I. Paul and Peter H. Brown. "Evaluation and Modification of CRAFT for an Architectural Methodology." In Emerging Methods in Environmental Design and Planning, ed. Gary T. Moore, 155-161. Cambridge: MIT Press, 1970.

Liggett, Robin S. "Floor Plan Layout by Implicit Enumeration." In Proceedings of the EDRA3/AR8 Conference held in Los Angeles, 1972. edited by W.J. Mitchell. 1972.

Liggett, Robin S. and William J. Mitchell. "Optimal Space Planning in Practice." Computer Aided Design 13, Number 5 (1981): 277-287.

Mackenzie, C.A. "Inferring Relational Design Grammars." Environment and Planning B 16, Number 3 (1989): 253-287.

Magee, Kevin. "The Elicitation of Knowledge From Designers." Design Studies 8, Number 2 (1987): 62-68.

Maher, Mary Lou. "Process Models for Design Synthesis." The AI Magazine 11, Number 4 (1990): 49-58.

Majkowski, Bruce R. and Yehuda E. Kalay. "Computability of Design." In Computability of Design, ed. Yehuda E. Kalay, 349-357. New York: John Wiley and Sons, 1987.

March, Lionel and J. Philip Steadman. The Geometry of the Environment. London: RIBA Publications Limited, 1971.

McBrien, A., Jim Madden, and Nigel Shadbolt. "Artificial Intelligence Methods in Process Plant Layout." In Proceedings of the 2nd International Conference IEA/AIE-89 held in Tullahoma, USA June 1989, 1989.

McDermott, John. "Domain Knowledge and the Design Process." Design Studies 3, Number 1 (1982): 31-36.

Minsky, Marvin. The Society of Mind. New York: Simon and Schuster, 1986.

Mitchell, William J. Computer-Aided Architectural Design. New York: Petrocelli / Charter, 1977.

Mitchell, William J. "Reasoning About Form and Function." In Computability of Design, ed. Yehuda E. Kalay, 89-97. New York: John Wiley and Sons, 1987.

Mitchell, William J., J. Philip Steadman, and Robin S. Liggett. "Synthesis and Optimization of Small Rectangular Floor Plans." Environment and Planning B 3, Number 1 (1976): 37-70.

Montreuil, B. "Requirements for Representation of Domain Knowledge in Intelligent Environments for Layout Design." Computer Aided Design 22, Number 2 (1990): 97-108.

Nii, H. Penny. "Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures." The AI Magazine (Summer 1986): 38-53.

Oxman, Rivka. "Prior Knowledge in Design: A Dynamic Knowledge-Based Model of Design and Creativity." Design Studies 11, Number 1 (1990): 17-27.

Oxman, Rivka, and John S. Gero. "Designing by Prototype Refinement in Architecture." In Artificial Intelligence in Engineering Design, ed. John S. Gero, 395-411. Southampton: Computational Mechanics, 1988.

Pascall, J.R., and G. Hamilton. "The Development of Expert Systems for the Design of Ventilation Systems in Office Spaces." To appear in Proceedings of the Symposium on Artificial Intelligence in Building Design: Progress and Promise held in St. Louis, June 1990, by the American Society of Heating and Refrigeration Engineers, 1990.

Pfefferkorn, Charles E. "The Design Problem Solver: A System for Designing Equipment or Furniture Layouts." In Spatial Synthesis in Computer-Aided Building Design, ed. Charles M. Eastman, 98-146. New York: Halsted Press, 1975.

Pohl, Jens, Art Chapman, Laurian Chirica, and Leonard Myers. "ICADS: An Intelligent Computer-Aided Design System." Design Methods and Theories 22, Number 1 (1988): 749-794.

Pohl, Jens, Art Chapman, Laurian Chirica, Bob Howell and Leonard Myers. Implementation Strategies for a Prototype ICADS Working Model. CAD Research Unit, School of Architecture and Environmental Design, California Polytechnic State University, San Luis Obispo, 1988. CADRU-02-88.

Pohl, Jens, Leonard Myers, Art Chapman, and John Cotton. ICADS: Working Model Version 1. CAD Research Unit, School of Architecture and Environmental Design, California Polytechnic State University, San Luis Obispo, 1989. CADRU-03-89.

Pohl, Jens, Leonard Myers, Art Chapman, James Snyder, Hannah Chauvet, John Cotton, Coe Johnson, and Dean Johnson. ICADS Working Model Version 2 and Future Directions. CAD Research Unit, School of Architecture and Environmental Design, California Polytechnic State University, San Luis Obispo, 1991. CADRU-05-91.

Radford, Anthony D. and John S. Gero. "Towards Generative Expert Systems for Architectural Detailing." Computer Aided Design 17, Number 9 (1985): 428-435.

Rehak, Daniel R. and H. Craig Howard. "Interfacing Expert Systems with Design Databases in Integrated CAD Systems." Computer Aided Design 17, Number 9 (1985): 443-454.

Rinsma, I. "Rectangular and Orthogonal Floorplans with Required Room Areas and Tree Adjacency." Environment and Planning B 15, Number 1 (1988): 111-118.

Rinsma, I., J.W. Giffin, and D.F. Robinson. "Orthogonal Floorplans from Maximal Planar Graphs." Environment and Planning B 17, Number 1 (1990): 57-71.

Rosenman, Michael A., John S. Gero, Peter J. Hutchinson, and Rivka Oxman. "Expert Systems Applications in Computer-Aided Design." Computer Aided Design 18, Number 10 (1986): 546-551.

Roth, J. and R. Hashimshony. "Algorithmis in Graph Theory and Their Use for Solving Problems in Architectural Design." Computer Aided Design 20, Number 7 (1988): 373-381.

Rychener, Michael D. Expert Systems for Engineering Design. Boston: Academic Press, Inc., 1988.

Schmitt, Gerhard. MicroComputer Aided Design for Architects and Designers. New York: John Wiley and Sons, 1988a.

Schmitt, Gerhard. "ARCHPLAN: An Architectural Planning Front End to Engineering Design Expert Systems." In Expert Systems for Engineering Design, ed. Michael D. Rychener, 257-277. Boston: Academic Press, Inc., 1988b.

Seehof, J.M. and W.O. Evans. "Automated Layout Design Program." Journal of Indutrial Engineering 18, Number 12 (1967): 690-695.

Shapiro, Stuart C. and James Geller. "Artificial Intelligence and Automated Design." In Computability of Design, ed. Yehuda E. Kalay, 173-185. New York: John Wiley and Sons, 1987.

Simon, Herbert A. "Style in Design." In Spatial Synthesis in Computer-Aided Building Design, ed. Charles M. Eastman, 287-309. New York: Halsted Press, 1975.

Simon, Herbert A. The Sciences of the Artificial. Cambridge, MA.: MIT Press, 1981.

Smithers, Tim. "AI-Based Design Versus Geometry-Based Design or Why Design Cannot be Supported by Geometry Alone." Computer Aided Design 21, Number 3 (1989): 141-150.

Sriram, D., R.Logcher, A. Wong, and S. Ahmed. A Case Study in Computer-Aided Cooperative Product Development. Intelligent Engineering Systems Laboratory, Department of Civil Engineering, Massachusetts Institute of Technology, 1990. IESL-90-01.

Steadman, Philip. "Graph-Theoretic Representation of Architectural Arrangement." In The Architecture of Form ed. Lionel March, 94-115. Cambridge: Cambridge University Press, 1976.

Stiny, G. "Introduction to Shape and Shape Grammars." Environment and Planning B 7, Number 3 (1980): 343-351.

Stiny, G. "Computing with Form and Meaning in Architecture." Journal of Architectural Education 39, Number 1 (1985): 7-19.

Stiny, G. "What is a Design." Environment and Planning B 17, Number 1 (1990): 97-103.

Teague, Lavette C., Jr. "Network Models of Configurations of Rectangular Parallelepipeds." In Emerging Methods in Environmental Design and Planning, ed. Gary T. Moore, 162-169. Cambridge: MIT Press, 1970.

Tommelein, Iris D. "SightPlan: An Expert System that Models and Augments Human Decision-Making for Designing Construction Site Layouts." Ph.D. diss., Stanford University, 1989.

Wade, John. "Any CAD Program Models (And Becomes) A Design Process." In Proceedings of the 1987 Conference on Planning and Design in Architecture Held in Boston 17-20 August 1987, edited by J.P. Protzen, 47-52. New York: The American Society of Mechanical Engineers, 1987.

Wade, John. The Fine Tools Project. College of Architecture and Urban Studies, Virginia Polytechnic Institute and State University, October 1989.

Weinzapfel, Guy and Steven Handel. "IMAGE: Computer Assistant for Architectural Design." In Spatial Synthesis in Computer-Aided Building Design, ed. Charles M. Eastman, 61-89. New York: Halsted Press, 1975.

Whitefield, Andy and Clive Warren. "A Blackboard Framework for Modelling Designers' Behaviour." Design Studies 10, Number 3 (1989): 179-187.

Willoughby, Tim M. "Building Forms and Circulation Patterns." Environment and Planning B 2, Number 1 (1975): 59-87.

Woodbury, Robert F. "Strategies for Interactive Design Systems." In Computability of Design, ed. Yehuda E. Kalay, 11-35. New York: John Wiley and Sons, 1987.