National Library of Canada
Bibliothèque nationale du Canada

Canadian Theses Service
Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

NL-339 (r. 88/04) c

Knowledge-Based System Approach to Building Envelope Design

Krishnan Gowri

A Thesis
in
The Centre for Building Studies

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montreal, Quebec, Canada

October 1990

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

ISBN 0-315-64721-3

ABSTRACT

KNOWLEDGE-BASED SYSTEM APPROACH TO BUILDING ENVELOPE DESIGN

Krishnan Gowri, Ph. D.
Concordia University, 1990

The successful performance of a building envelope assembly depends on how well the individual components are designed and integrated together in meeting the objectives of moisture protection, energy efficiency, thermal comfort, fire and acoustic resistances, and cost. Decisions on materials and construction types for envelope components are often made during the preliminary stages of design when the available information is limited to building geometry, user and owner requirements. Lack of information, time pressures, and the need to consider a large number of performance attributes make it difficult for the designer to examine all the possible design alternatives. This is a problem of information processing and decision making which are difficult to solve manually.

The present study aims to improve the building envelope design process by developing a systematic approach for synthesis and evaluation of alternatives at the preliminary design stage. Knowledge-based system techniques are utilized to automate the information processing and decision making problems in design. It is possible to develop a knowledge base consisting of building Code requirements, performance standards, material properties, construction types and design heuristics. Such a knowledge base can provide the information necessary to establish the performance requirements of the design context and to generate design alternatives. A hierarchy of building envelope components and materials with their semantic relationships and functional attributes is developed using a schema (frame) based knowledge representation technique. Generation of feasible design alternatives is viewed as a constraint-based search problem in which the envelope components individually and collectively must satisfy the performance requirements of the design context. Problem decomposition and multiple levels of generate-test paradigms are used in the alternative generation process. Ranking and selection of alternatives can then be accomplished by specifying the preferences on construction types and the priorities on performance attributes.

The above methodology has been implemented in a software prototype known as Building Envelope Analysis and Design System (BEADS). The prototype knowledge base includes information from building code, a performance standard, design manuals, practising architects, material properties and cost data handbooks. A design query language (DQL) has been developed to facilitate rapid design revisions, re-evaluation of alternatives and to maintain the knowledge base. DQL provides designer freedom and allows the maintenance of knowledge base without having to familiarize with the development environment. BEADS currently addresses only a few important performance requirements but demonstrates the advantages of knowledge-based system approach. Two sample design problems have been solved to validate the knowledge base and to identify future extensions.

It is shown that knowledge-based system approach to building envelope design can significantly improve the abilities of the designer to consider a large number of alternatives and many performance attributes simultaneously at the preliminary design stage. The knowledge base can also serve as a vehicle for technology transfer in the building industry. The knowledge representation scheme and inference mechanisms developed for the BEADS prototype can be adapted to provide an efficient methodology to automate other formulation type design problems.

To all my teachers

ACKNOWLEDGEMENT

I wish to express my gratitude and appreciation to Dr. Paul Fazio for initiating this research and for his valuable guidance, criticisms and suggestions throughout this study. This work would not have been possible but for his continued interest in innovative approaches to integrated building design.

I am thankful to Dr. Claude Bedard for the many helpful suggestions and discussions, and interest shown during the course of this study.

Architects Mark Benoit, Pierre Collette and David Reich are gratefully acknowledged for their assistance in reviewing the knowledge base and for providing the test problems. Additional thanks are due to the faculty and students at the Centre for Building Studies and to my colleagues at SIRICON.

Finally, my sincere thanks to SIRICON for providing the computer resources, time and encouragement to carryout this study.

TABLE OF CONTENTS

4. Knowledge Representation

5. Implementation and Validation

6. Conclusions and Recommendations

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 General

The building industry is facing a problem of predicting quality and performance of constructed facilities. Increasingly a large number of court cases are reported relating to the poor quality of construction, inadequate performance and cost over-runs, etc. ([1], [2], [3], [4]). Such instances are frequently attributed to the lack of information and communication among professionals responsible for design. Decisions made during the preliminary design stages greatly influence the successful realization of a building project. The growing complexity and interdisciplinary nature of building design requires an integrated approach involving various disciplines e.g. building science, structures, mechanical, electrical and other specialties. It is often difficult for designers to consider the interplay of the many disciplines in sufficient detail, particularly in the conceptual stages. Automated design aids are vital to the decision making process in the preliminary stages of design.

1.2 Building Envelope Design Issues

As soon as a building project is conceived by an owner/developer, an architect is consulted for developing the concept. In the earliest stages of design, schematic drawings essentially consist of floor plans and elevations. During this stage, decisions are made regarding the building type, occupancy, form and fabric, enclosure dimensions and materials, etc. Any decision made at this preliminary design stage will be reflected on the quality, performance and cost of construction and maintenance of the building.

In North American climatic conditions, the building envelope plays a very important role in protecting the indoor environment from the rapidly fluctuating outdoor weather conditions. The material and construction cost of the building envelope may be from 10 to 20% of the total building cost and it has a significant impact on the cost of energy for the heating and cooling of the building. Typically, building envelope components must be designed to meet a number of performance requirements relating to thermal, moisture protection, structural safety, fire resistance, acoustic resistance and aesthetic characteristics. In order to satisfy these requirements, many different materials with specific performance characteristics are often combined to form an assembly. Such assemblies once developed and successfully used are treated as standard assemblies for future use by designers. In a study by Mackinder [5], it has been found that many designers have preferences for materials and components backed by an argument based on experience of previous use to fulfill a particular set of criteria which also happens to be appropriate to the current design situation. But it is not always assured that design contexts will be exactly the same. Hence unforeseen problems are often encountered during construction or later during the life of the building.

There are many cases of construction claims and lawsuits where problems not anticipated at the design stage resulted in major cost over-runs, redesign of certain building component details causing conflicts among the trades and delaying the construction time. Fazio [1] reports of a building envelope problem in which the interaction between the cladding and structural systems was not properly addressed in the design stage. Low tolerances were assumed for the steel structure which were not practical causing difficulties during installation and resulted in construction delay requiring the redesign of envelope components. The contractors claim for six million dollars for cost over-run due to conflicts in design were settled out of court at two million dollars. This case clearly demonstrates the lack of integration and the need to address the interaction between various sub systems during the design process. The Alberta Building Envelope Council presents several such cases of problems ([2],[3],[4]) where integration issues are found to be the predominant cause of envelope related failures. Material incompatibilities, missing air-vapour barriers and improper location of insulation and other materials in the envelope assembly are some of the basic building envelope problems in design.

Development in new constructional types and more competitive materials appear on the market phasing out the traditional ones. This is a problem of concern to the designer, as the use of new materials and systems requires a thorough investigation which may not be possible during early stages of design. Hence the chances of using a more efficient and economical material or system may not be feasible due to the lack of supporting evidence. In search of an optimally best performance for the envelope, the designer needs to consider all feasible combinations of materials and constructional types. The number of available materials, performance attributes and alternative combinations is very large; and this leads to a problem of information processing and decision making which are difficult to solve manually.

1.3 Computer Aided Building Design

Computer aids have been developed for assistance in analysis, design, drafting, project management, cost and quantity estimation, and many other tasks in the building construction process. But almost all the efforts so far have been to use the computer as a tool for solving numerical problems, drafting and report preparation. The impact of computers in architectural design has been marginal with more than 90% used in drafting [6].

Design in general is an iterative process in which the sequence of operations depend on the experience, knowledge and imagination of the designer [7]. The information required in the design process is not always complete and many possible yet acceptable solutions may be obtained for the same problem. Hence all feasible design solutions need to be considered to identify the most appropriate one. This is a critical issue in building design in which each design is unique and each project has specific requirements. According to Kalay [6]:

> *" No model has been found which encompasses in a single comprehensive manner both the internal relationships between the numerous components of a building, and the conditions that are external to it (e.g: environmental, social, psychological, etc.). Without such a comprehensive model, analytical processes that rely on it as a source of information are found to be limited in their scope, and unable to take into account the many factors and the trade-offs that affect certain conditions. The lack of adequate analysis, in turn, hinders the generation of new states that could advance the process towards successful completion."*

Thus the role of computers in building design should be viewed as information processing tools providing the flexibility to experiment with design concepts rather than just performing numerical computations.

Recent progress in artificial intelligence and knowledge-based system techniques shows that some of the problems in automating the building design process can now be addressed to develop tools for use at the preliminary design stage. Symbolic and explicit representation of design objects, separation of domain knowledge from inference mechanism, and the ability to incorporate different problem solving techniques are the advantages of using knowledge-based systems approach to design. Many prototype knowledge-based systems have been developed for structural design, code compliance checking and other design problems. Application of the knowledge-based system approach to building envelope design will provide a framework for integrating the various types of information such as performance requirements, design heuristics, code requirements, material properties, constructional types and the flexibility to implement search strategies for synthesis and evaluation of design alternatives.

1.4 Research Objective

There is a need to develop models and tools for early design stages because of the high impact of early design decisions on the building quality and performance [8]. Selection of materials and constructional types for the building envelope are among the most important decisions made during the preliminary design stage. Little work has been done so far to address the problems in identifying appropriate materials and constructional types for the building envelope components by considering many possible design alternatives and performance attributes simultaneously. Hence the objective of the present study is to develop a systematic methodology to formalize the building envelope design process for synthesis and evaluation of design alternatives and to simultaneously consider the many aspects of performance.

The proposed methodology is intended for use at the preliminary design stage and it is assumed that only basic decisions on building form, location and dimensions are available. Knowledge-based system approach is identified as suitable for representing the knowledge and information required in developing the design from user requirements to object descriptions of building envelope components. Designer freedom for expressing the preference on materials, constructional types and performance attributes is considered to be an important part of the design process. Design manuals, code specifications, performance standards and practising architects have been consulted for developing a knowledge base to address some of the most important aspects of building envelope design. Available building materials, their properties, details of constructional types are represented in a hierarchy of objects and relational links in a data base.

A prototype implementation of a Building Envelope Analysis and Design System (BEADS) is attempted to demonstrate the practicality and advantages of the proposed methodology. The BEADS implementation includes the development of a knowledge base, design alternative generator and evaluator, and a design query language user-interface. A few test problems have been solved to validate the BEADS knowledge base and to identify future research issues.

1.5 Organization of the Thesis

The next chapter presents a review of literature in building envelope design, previous work on design models and discusses the proposed design methodology..

Chapter 3 discusses the knowledge-based systems approach to design problems and presents an evaluation of available knowledge-based system tools for use in the present study.

Chapter 4 deals with the knowledge acquisition and representation of the design context, design constraints, material and constructional types information in the BEADS knowledge base. The design alternative generation and evaluation methodologies are also described.

Chapter 5 provides the details of BEADS implementation and test problems. The development strategy for the constraint-based design alternative generator and design query language user-interface are discussed. Details of test problems and the validation process are also presented in this chapter.

Chapter 6 concludes by identifying the contributions of this study and describes the significance of the research outcome. A number of research issues for extending the present work and to develop practical solutions to the building envelope design problems are also presented.

CHAPTER 2

APPROACHES TO BUILDING ENVELOPE DESIGN

2.1 Introduction

The realization of a building from conception to commissioning goes through a number of stages including preliminary design, detailed design and construction. Figure 2.1 shows the influence of decisions made at the various stages on the life cycle cost of the building [9]. It is evident that the preliminary design stage is critical for ensuring successful performance, high quality and reduced cost. Major decisions on building form, dimensions and envelope components are made during this early design phase. Traditionally, cost or aesthetic values were the prime concerns in selecting materials and designing envelope systems. But the energy crisis of the 70's caught the attention of designers to improve the thermal performance of the building envelope. Consequently problems of high indoor humidity levels, air leakage and indoor air quality need to be addressed in the design. Thus building envelope design becomes complex, trying to resolve the conflict among performance attributes and to identify the best combination of materials and systems. As the number of attributes and alternative combinations are large, it is necessary to develop a systematic methodology for exploring the design space. The following sections briefly describe the functions of the building envelope, decisions made in the design process and a review of previous work. A knowledge-based system approach to solve some of the problems in building envelope design is proposed by developing a framework for systematically exploring the feasible alternatives and in making knowledgeable design decisions.

2.2 Role of Building Envelope

The primary function of a building envelope is to separate and maintain the living conditions of an indoor environment, preventing it from being affected by the rapid variations in the outdoor weather conditions. Exterior walls, roof, windows, doors, basement walls and floor are the major components of building envelope assembly responsible for protecting the indoor environment. The degree of protection offered by

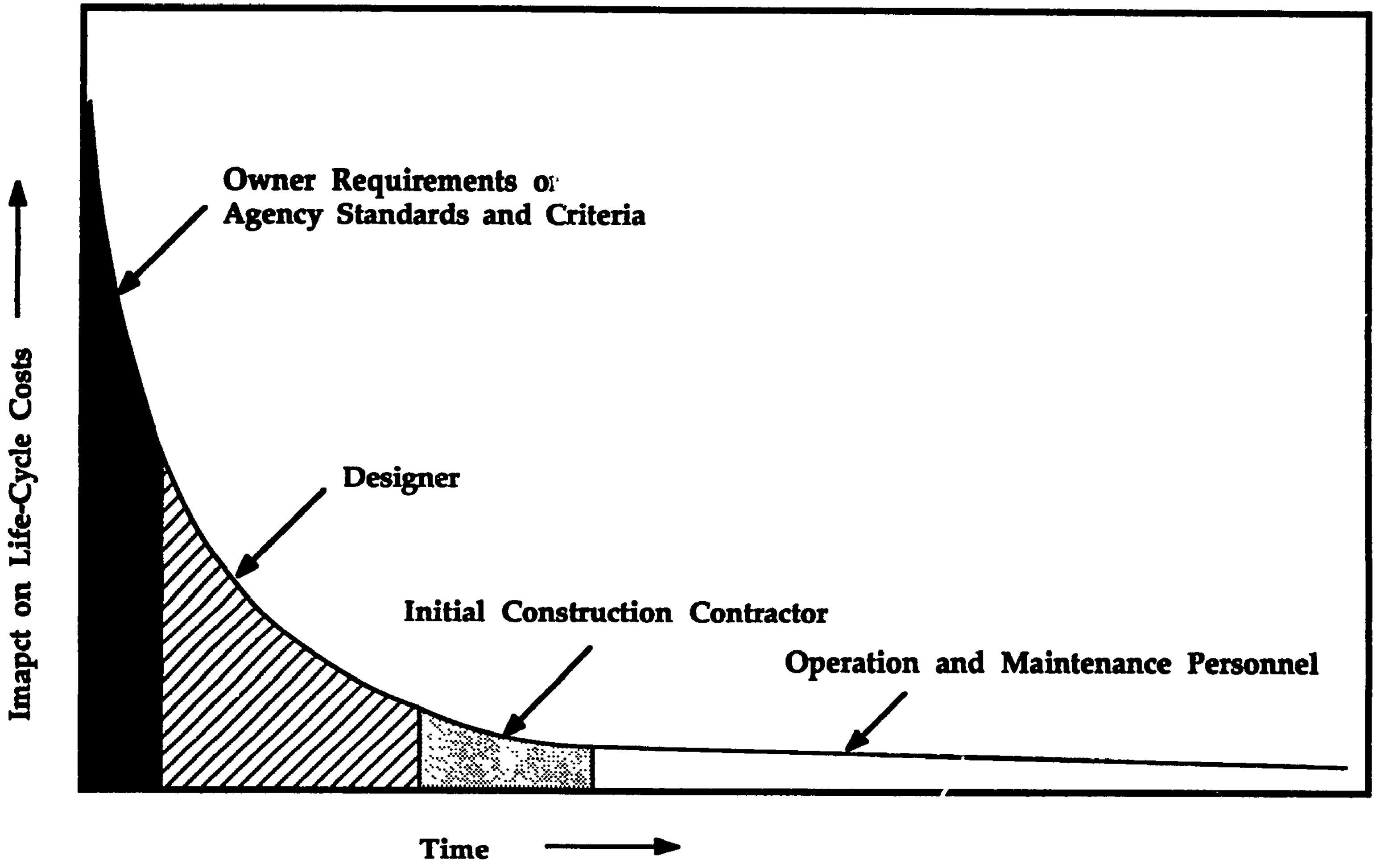


Figure 2.1: Impact of Major Decision Makers on Building Life-Cycle Costs [9]

the building envelope is dependent on the design objectives in terms of energy efficiency, cost, durability and other performance attributes. A list of principal requirements to be considered in the design of external walls [10], extended from that suggested by Hutcheon [11] is given below:

(i) Control heat flow
(ii) Control air flow
(iii) Control water vapour flow
(iv) Control rain penetration
(v) Control light, solar and other radiation
(vi) Control noise
(vii) Control fire
(viii) Structural safety (strength and rigidity)
(ix) Durability
(x) Aesthetic quality
(xi) Cost
(xii) Buildability
(xiii) Maintainability
(xiv) Special requirements.

These requirements though originally developed for external walls, are applicable to other components of the envelope as well. Traditionally design of walls has been based on structural load carrying capacity, cost and aesthetic value. The demand for increased quality and performance requires that all the above aspects of the envelope be considered in design. This is not a simple task because of the conflicting and complex relationship between the performance attributes. For example, an insulation material with good thermal resistance characteristics may not be offering the degree of moisture resistance to avoid interstitial condensation. Hence the selection of materials and the environment in which they are to perform must be carefully established. As there may not be any one single material with the desired level of performance to meet all the above requirements, many different materials are combined together as an assembly. Typically external walls consist of various layers such as cladding, insulation, structure, air-vapour barrier and interior finishing. External cladding often serves as a barrier for excluding bulk moisture transport and adds to the aesthetic value of the building facade. The thermal efficiency of the wall is dependent on the properties of the insulation layer. The air-vapour barrier provides resistance to moisture diffusion due to exfiltration/infiltration and high humidity levels, and minimize air leakage. Structural framing or a load bearing wall offers structural support for the wall assembly and transfers the wind load on cladding to the foundation. Gypsum wall boards for fire resistance and interior finishes to suit user needs should also be provided. The location of insulation and air-vapour barrier layers must be carefully considered during design. Materials for each layer should be selected to meet both their individual requirements and collectively as a system to ensure compatibility and successful performance. Windows and doors in the external walls are important for providing natural lighting, occupant comfort and safety. Thermal resistance, shading factors and strength of glazings have to be considered in design. Integration of windows and doors in the envelope assembly is critical in avoiding leakage paths and thermal bridges, and ensuring proper functional performance.

Roof systems consist of deck, sheathing, air-vapour barrier, insulation, water proofing membrane and surfacing materials. Resistance to snow loads, rain water, thermal delamination and durability of water proofing membrane require special attention in the design of roof systems. The continuity of air-vapour barrier between roof and wall systems is essential for the successful performance of the envelope assembly. More detailed discussion on the functions of building envelope components and building science principles for design can be found in [12], [13], [14], [15] and [16].

2.3 Decisions in Building Envelope Design

As discussed earlier, the degree of separation offered by the building envelope is a design decision which influences the selection of materials and constructional types. Design objectives such as minimum cost, minimum energy consumption, good fire resistance, minimum sound transmission and durability must often be considered simultaneously, thus making the selection process more difficult.

Figure 2.2 shows a typical sequence of building envelope design decisions as described in the Architects' Journal handbook of building enclosure [12]. Identification of user needs and subsequently the establishment of performance requirements are fundamental to the design process. Decisions on material selection and constructional details are then made to meet the design objectives. Mattar [17] identifies the building envelope design process to consist of the following stages:

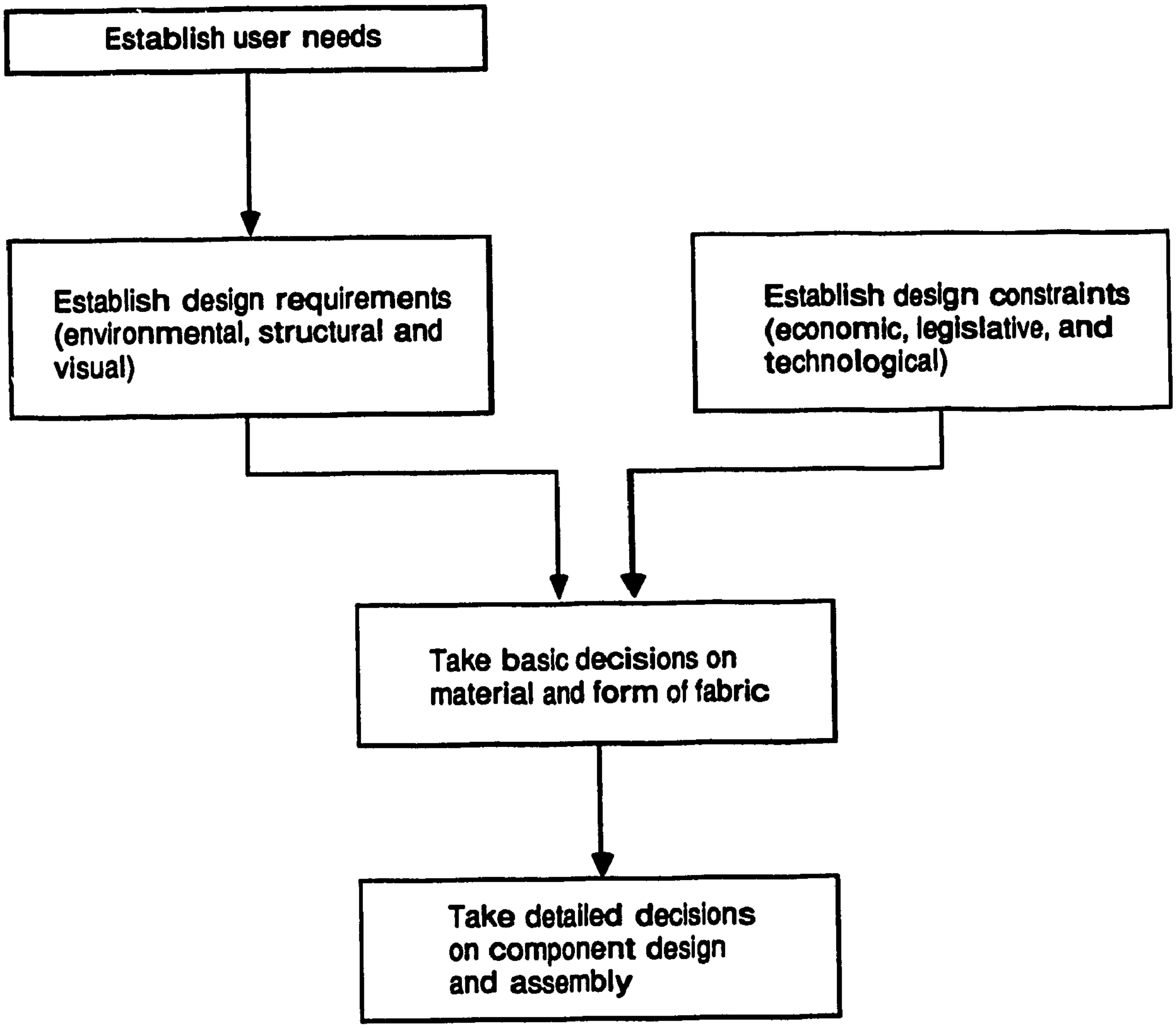


Figure 2.2: Sequence of Design Decisions [12]

(i) Establishing the design context
(ii) Specification of performance objectives
(iii) Generation of design alternatives
(iv) Prediction of alternative performances
(v) Evaluation and selection of a suitable alternative

In each of the above stages, the designer typically makes decisions which become constraints for defining the boundaries of the design search space and in this process eliminate unacceptable alternatives. The design context describes the owner and user requirements of the building with regard to location, type of occupancy and geometry, etc. These details can be used for specifying performance objectives for the envelope components and assembly in quantitative terms, the thermal resistance, structural strength, fire resistance, acoustic resistance and others as required for design.

The more the number of performance attributes considered at this stage, the more complex the design process will be, and consequently, the outcome will be more reliable. Design alternative combinations of materials and components are often provided by manufacturers and design handbooks. Designers also synthesize new alternatives based upon need and subject to availability of resources. Such designs must be analyzed to predict their performance and checked for satisfactory compliance with the requirements developed in stage (ii). Once when all possible alternatives are considered and their predicted performances are known, they now can be compared and evaluated based on priorities of performance. During this stage many decision making methodologies can be used in assisting the designer to select the best suitable alternative. Some of the available methods are reviewed in the following section.

2.4 Review of Design Research

Building envelope design research is interdisciplinary in nature and brings together many disciplines such as design methodologies, materials science, building physics, heating, ventilating and air-conditioning systems, construction technology, mathematical optimization, decision making approaches and information processing. This section briefly presents a review of previous work on design methodologies and operation research techniques aimed at developing a systematic approach for automating the design process in general and building envelope design in particular.

2.4.1 Design Methodologies

Design is a purposeful behaviour which is directed at devising artifacts or environments that attain certain goals while abiding by certain constraints [6]. Since no formula exists which can translate goals and constraints into a self-consistent physical form, design is an interactive, educated "trial-and-error" process that relies heavily on knowledge and experience. Logcher [7] defines engineering design as an iterative process in which the sequence of operations depends on the nature of problems and on the experience, knowledge and imagination of the designer.

Many researchers from different domains have attempted to develop methods and models for design. There is no general consensus on what design is and how this is carried out by individuals. A review of design research by Finger and Dixon [18] in the field of mechanical engineering presents a classification of design models as:

(i) Descriptive models
(ii) Prescriptive models
(iii) Computer-based models

Descriptive models are derived from protocol analysis, cognitive science and observation of design process from case studies. Prescriptive models are developed based on identifying the various stages in design process and the attributes for defining the design artifact. Design process is an interactive progression through the stages of recognition of need, specification of requirements, concept formulation and selection, embodiment of design detail, and production and maintenance. Detailed methodologies for each stage can be developed for design. Computer-based models are aimed at developing tools to assist in the design process. Parametric design, configuration design and conceptual design are the three most popular categories of computer-based models. In parametric design, values are assigned to the attributes of an artifact which are design variables. Configuration design aims at transforming a physical concept to a configuration with a defined set of attributes. This is achieved by assembling a set of standard components or by redesigning non-standard components to meet the functional requirements. Conceptual design is used for generating design descriptions from functional or behavioural requirements. Much of the research in computer-based models is due to the tools and techniques resulting from developments in the area of artificial intelligence.

Design problems are in general ill-structured and based on trial and error in finding a solution. Architectural design situations radically differ from algorithmic and mathematical procedures as they traditionally consist of arranging volumes in space, selecting materials, constructional and environmental systems and require only a small amount of computation [19]. Architectural design has no developed and generally accepted theories of how to produce architectural design [20]. Often experiences in problem solving is used for design. This experience may be in the form of similar buildings completed by the practice, or in the form of guides, standards or articles in the technical press. These general solutions then provide a starting point from which an architect explores the characteristics of the particular context of design. The simplest model of architectural and engineering design process consists of three interrelated subprocesses of definition, synthesis-analysis, and documentation as shown in Figure 2.3 [21]. Definition refers to the identification of a given need and to a thorough specification of the object to be designed. This specification includes human, functional and physical characteristics or constraints, all of which incur cost consequences [22]. Synthesis and analysis are closely related and highly iterative. Components or sub-systems are first put together, then subjected to performance evaluation and compared against criteria set earlier, finally modified and re-evaluated. This subprocess is repeated until all components and sub-systems are optimized to form an overall system [23]. Documentation is finally concerned with the production of drawings and written specifications sufficient to realize the given object.

Building envelope design as performed at the preliminary design stage can be viewed as a search process to identify the suitable combination of materials and construction types satisfying the various performance constraints. This exploratory model for design is necessary to overcome the difficulty of addressing the many different and conflicting performance requirements of the sought environment or artifact into a self-consistent physical or organizational assembly of individual components. A comparative evaluation by Kalay et al [24] of the various search techniques for modelling the design process shows that constraint-based techniques are best suited for synthesis - evaluate problem solving. Gross et al [25] have discussed the use of "Constraint Models" for demonstrating the computability of design. Chan and Paulson [26] have shown that the explicit representation of constraints can be used in exploratory design to devise design descriptions in a way that guarantees the result to be consistent with constraints instead of using constraints to merely check design descriptions.

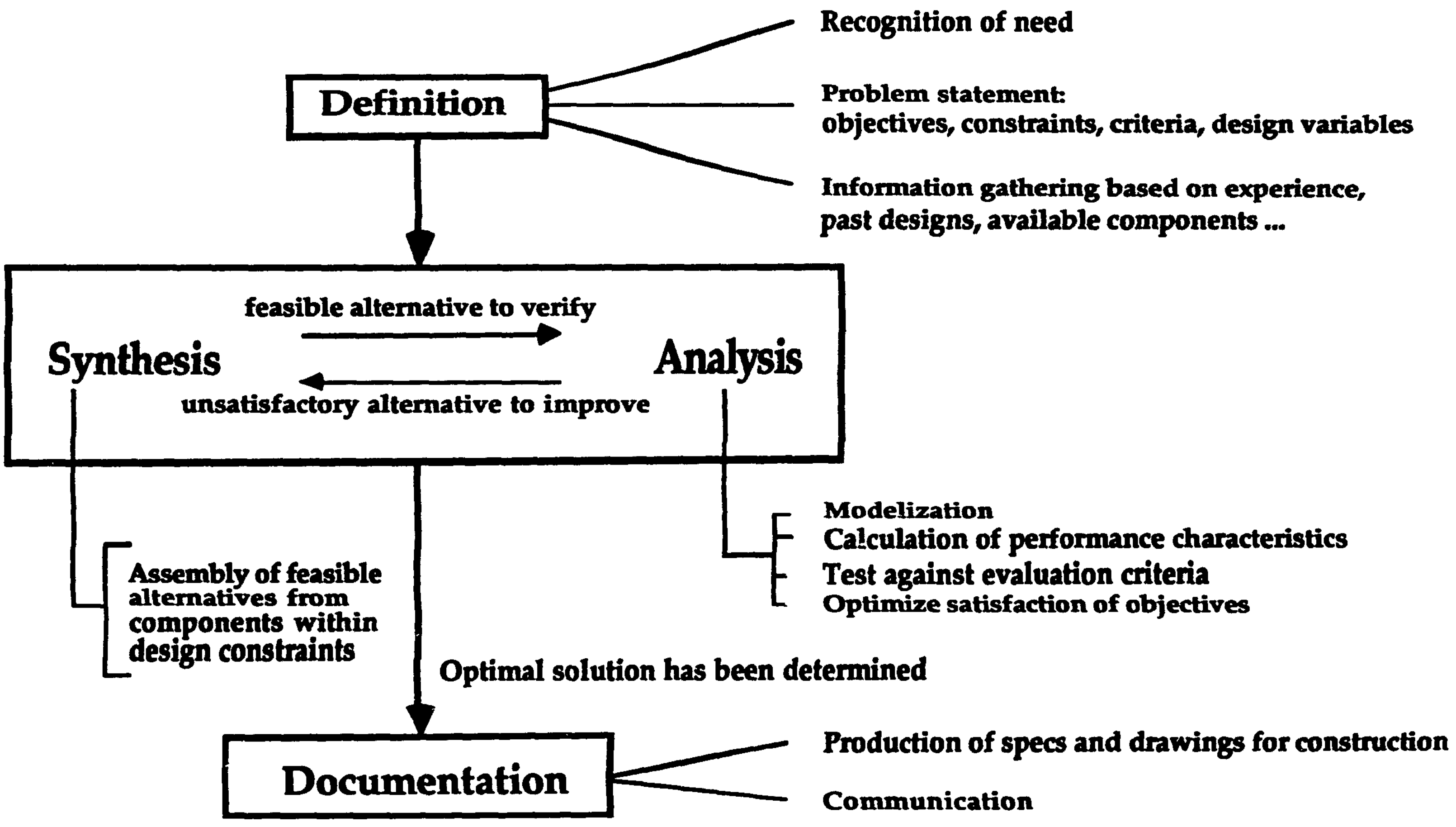


Figure 2.3: Model of Building Design Process [21]

There have been no specific attempt to model the building envelope design process excepting for the sequence of design decisions and the various stages of design discussed in the previous section. The above review of design methodologies shows that the exploratory model of constraint-based search can be adapted for automating the building envelope design process.

2.4.2 Operations Research Techniques

Research efforts to develop design aids for building envelope have applied decision theories and optimization techniques. The objective of using operation research techniques is to identify the best design solution by considering a number of feasible alternatives and a set of performance attributes. Mattar [17] presents a "Decision by Exclusion" methodology based on utility theory principles for finding the optimum design alternative from a set of feasible ones. Five performance attributes have been considered in this design problem. These attributes correspond to fire and acoustic resistances, thermal resistance, risk of condensation, sound transmission coefficient and initial cost of the assembly. The predicted behaviour for each attribute for all the alternatives are calculated on a normalized percentile scale. Then the decision process involves three steps as below:

(i) Exclusion of dominated alternatives - without consideration of priorities among objectives
(ii) Further exclusion of alternatives by specifying priorities among objectives.
(iii) Robustness analysis.

Designer preferences are expressed as utility functions using a set of partially ordered priorities among objectives and these are applied simultaneously to eliminate the undesirable alternatives. Then an additive utility model is used for ranking the feasible solutions. Robustness analysis is provided as an alternate method for ranking performance attributes instead of the single weight used in additive utility models.

Mattar has shown that the preferences among performance attributes and the level of performance expected for each attribute have a great impact on the design decisions, whereas the use of different utility functions has only a marginal effect in choosing the the best alternative. Mattar's study assumes that the first four stages of design process are completed and all the information for design context, generation of alternatives and prediction of their performances are available. In practice, this is not possible, as it requires an extensive amount of information, knowledge and experience. Hence the application of the decision methodology is limited by the ability to define the performance requirements of the design context, the identification of all possible design alternatives and predicting the performance of these alternatives. The magnitude of information handling problem is greatly accentuated by the number of performance attributes and the number of envelope components to be considered. For example, if the designer wants to find an optimum combination of walls, roof, windows and doors, then the solution space becomes very large and a more systematic methodology is required.

Mathematical optimization techniques have been extensively used for material selection and building design problems ([27], [28]). Optimization is the process of finding the best solution to meet the objectives or goals of a design and subject to a set of constraints. Numerous mathematical formulations for optimization are available depending on the number of objectives and types of constraints. Radford and Gero [27] have reviewed the various optimization techniques and their application to building design problems. The following is a classification of optimization techniques commonly used in design problem solving:

(i) Linear programming
(ii) Non-linear programming
(iii) Dynamic programming

Linear and non-linear programming techniques are efficient methods to solve problems that can be modeled by linear and non-linear relationships respectively. Dynamic programming is used to solve large problems which may be decomposed to subproblems consisting of discrete, non-linear and discontinuous variables.

In another study, Murthy [29] utilizes the multicriteria optimization model to identify materials and material systems with multifunctional capabilities for the design of building enclosures. The multifunctional requirements considered are from environmental, structural, resource conservation and economic considerations. In an effort to explore the decision space thoroughly, the non-preference technique has been used. The performance attributes considered include thermal transmittance, time lag, attenuation factor, capital energy consumption, mass, overall thickness and overall cost for external wall systems. A two level optimization approach is adopted for selecting the alternatives and in evaluating them. The first level identifies the material systems with the specified multifunctional performance characteristics and the second level utilizes the material systems so identified to enclose a space and generates a Pareto optimal solution set with two dimensional trade-off diagrams for comparing the alternatives. Detailed analysis as part of the design is done in evaluating certain performance characteristics such as capital energy consumption, operational energy demand for heating and cooling, etc. The advantage of this approach is that the designer has complete freedom to specify the materials, their thicknesses and the location within the building element, and to explore the relative performance in making a selection. Since the optimization methodology is based on a non-preference technique, extensive number of combinations have to be analyzed, and the number of feasible alternatives generated increases exponentially with the increase in the number of different materials and layers specified. The two sample problems reported by Murthy show that there are 56 and 199 feasible alternatives possible for a three and four layer wall systems with 15 and 23 different materials respectively. It is suggested that the designer should use trade-off diagrams (one for each of the chosen performance attribute, with a standard axis of reference), for selecting a final design. Because of the level of detail and complexity in formulating the optimization problem and in analyzing the results, the above methodology is difficult to use in practice. Extensive computing facilities, and an understanding of optimization methodologies would also be required for applying this technique. Since the analysis and evaluation process are thoroughly treated, the solutions obtained would be mathematically reliable.

A more recent work by D'Cruz and Radford [30] uses a multicriteria model for assisting designers at the conceptual design stage. This model provides prescriptive quantitative information on the resolution of design conflicts between capital cost, thermal performance, planning efficiency and daylight availability in the choice of form and construction of paralellopiped open plan office buildings. Multicriteria Pareto optimal dynamic programming has been used in the evaluation of alternatives, with trade-off diagrams. Gero et al ([31], [32]) have extensively applied this methodology to solve building envelope design problems. The disadvantage of optimization lies in the inherent difficulty in formulating meaningful quantifiable objectives in building envelope design which is characterized by multiple and ill-defined objectives [33].

Even though the mathematical optimization techniques are useful in evaluating the alternatives, the difficulties in establishing the design context and performance attributes, and in generating the alternatives are not resolved and is expected of the designer, which is difficult to achieve in practice.

2.5 Knowledge-Based System Approach

Recent developments in artificial intelligence and knowledge representation techniques have provided the opportunity to develop computer programs that can emulate the human thought process. Traditional computer programs for design have relied upon algorithmic and procedural techniques essentially providing the facilities for number crunching. But most engineering design problems are not amenable to sequential processing and often require the use of design heuristics, compliance to codes and standards, etc. Knowledge-based system techniques can address some of these difficulties by enabling the representation of symbolic information and separation of domain knowledge from the inferencing process.

Research attempts so far have demonstrated the advantages of knowledge-based system application to preliminary design ([34], [35]) and code compliance checking problems ([36], [37]). The earliest work in developing knowledge-based systems for building design has focused on the selection of structural systems in the preliminary design stage. HI-RISE [34] and ALL-RISE [38] are two prototypes aimed at generating alternative structural systems for high rise buildings. Use of heuristic knowledge in the design process and the ability to develop design descriptions at the preliminary stage are the characteristic features of these implementations. Building design decisions at the preliminary stage are often more complicated because of the need to consider the influence of sub-systems such as the envelope, mechanical and electrical systems. Hence the integration of design and addressing the compatibility among the various building components is critical at the preliminary design stage. Integrated Building Design Environment (IBDE) is an effort recognizing one aspect of this issue and providing an integration platform for carrying out design from the preliminary to detailed design stages [39]. But a horizontal integration to consider more performance attributes and sub-systems needs to be solved. Building envelope design is representative of this situation in which the selection and design of materials and constructional systems are influenced by the type of structure, HVAC systems and many different performance requirements.

There are two prototype knowledge-based systems relevant to building envelope design that have been reported so far. The work by Bordeau [40] aims at determining the thermal characteristics of the envelope for meeting the heating system needs. This system considers the thermal conductivity of envelope components in isolation for the purposes of examining suitable insulation materials and does not consider many other envelope performance requirements. Tham, Lee and Gero [33] have demonstrated the concept of prototypes as applied to building envelope design. This system relies on the retrieval and refinement of prototype design situations for a given design context. The number of available prototypes in the knowledge base determines the applicability of this system in practice and the knowledge acquisition process is critical to develop prototype refinement strategies.

The objective of building envelope design systems in the preliminary stage is to assist the designer in establishing the expected level of performance and identifying alternative design solutions. Existing tools and techniques for envelope design are far from achieving this objective and lack the ability to consider many performance attributes and alternatives simultaneously. Any attempt to automate and formalize the design process should address the issues of information handling and decision making. The 'knowledge base' applicable to a particular building or product is seldom found in a single publication, and is more likely to comprise material published in text books, design guides, data books and data bases, codes of practice and technical regulations, etc., as shown in Figure 2.4 [41]. Building codes, performance standards, material properties, material compatibility information and design heuristics are typically the knowledge required in building envelope design. It is possible to develop a data base of material properties and constructional systems for the various envelope components, and integrate this with a knowledge-base of design heuristics and performance standards for the generation and evaluation of feasible alternatives.

USER NEEDS (ISO DP 6241)

Safety
Habitability
Suitability for use
Durability/reliability
Economy

CONTEXT (ISO DP 6241)

Climate
Site
Occupancy effects
Design consequences

BEHAVIOUR IN USE (CIB Master List)

Structural properties
Fire properties
Effects of gases, liquids and solids
Thermal properties
Acoustic properties
Optical properties
etc...

PREDICTIVE METHODS

Laboratory testing
Full-scale testing
Calculation
Conformity with designs known to be satisfactory

Figure 2.4: The Knowledge Base [41]

2.5.1 Proposed Methodology

The review of design methodologies and previous work on building envelope design ([12], [17], [18], [20], [21], [29]) shows that a systematic approach to automate the design process consists essentially of the following three stages:

(i) Establish the design context
(ii) Generate feasible alternatives
(iii) Rank and select alternatives

These three stages closely represent a model for systematic design as performed by designers in practice. A framework for integrating the above stages of design process will reduce most of the information handling and decision making problems encountered at the preliminary design stage.

The basic building data and user requirements can be used to establish the quantitative performance requirements of the design context. Performance requirements can be defined as technical statements developed from identified user needs and objectives that indicate an expected level of performance in order to fulfil a given function [42].

Building location and size, occupancy type, typical floor area and geometric data are available during the preliminary design stage. This data can be used to derive detailed functional performance requirements based on building codes and standards. For example, the ASHRAE Standard 90A-1980 [43] provides the minimum thermal performance requirements for the envelope components. Such an approach to establish the design context seems appropriate, particularly during the preliminary design stage, because performance requirements in general cannot always be determined with great precision due to deficiencies in understanding and lack of data [41]. Also even the best test methods are not so accurate as to justify the over-rigid boundaries between performance attributes. Any attempts to define the performance requirements must rely on experiential knowledge and many different information sources to obtain the necessary design data. Once defined, these performance requirements become constraints to identify feasible design alternatives which have satisfactory performance.

There are several possible wall, roof and glazing systems available in design handbooks and manufacturers literature. The designer needs to evaluate these systems and their performance for each of the attribute considered and determine their feasibility. This is not a simple task given the infinite number of alternatives possible by using different material and system configurations. An efficient way to address the issues of information handling at this stage would be to utilize constraint-based search techniques for identifying the feasible alternatives. In the context of the present study, constraint-based search can be viewed as a combination of problem decomposition and multiple generate-test paradigms. Design of building envelope components such as walls, roof and windows can be done separately to meet their respective performance requirements and then assembled together to satisfy the overall requirements of the design context. The major difference between the present approach and the previous attempts lies in this stage of generating feasible alternatives. Figure 2.5 shows a hierarchical decomposition of building envelope components and an example of external wall description in terms of basic constructional type and insulation type. At the lowest level of abstraction, building materials and their properties are sufficient to develop design descriptions using bottom-up generation techniques. But there are types that are already developed and found to perform successfully. A knowledge base consisting of this information and related design heuristics will considerably reduce the effort in generating new alternatives for building envelope components. For example, a basic wall type such as "Stucco on Sheathed Stud Wall" can be defined with the various layers of materials used and the location of insulation. During the generation process, it is possible to identify an appropriate insulation material such as rigid urethane board satisfying the performance requirements of the wall assembly. This approach to generation of design alternatives can be achieved only with the proper representation of materials, constructional types, their properties and functional relationships. The constraints defined by the performance requirements of the design context can be used to check each design alternative being generated both at the component level and at the system level.

The set of feasible alternatives thus generated must be ranked according to their relative performances before a decision can be made on selecting the most suitable one. Mackinder [44] reports on nine methods of decision-making techniques commonly used in architectural practice. There is no easy way to assess which one of these methods is best suited for building envelope design alternative selection. The main problem in assessing the available methods of decision making is due to the difficulty in formalizing

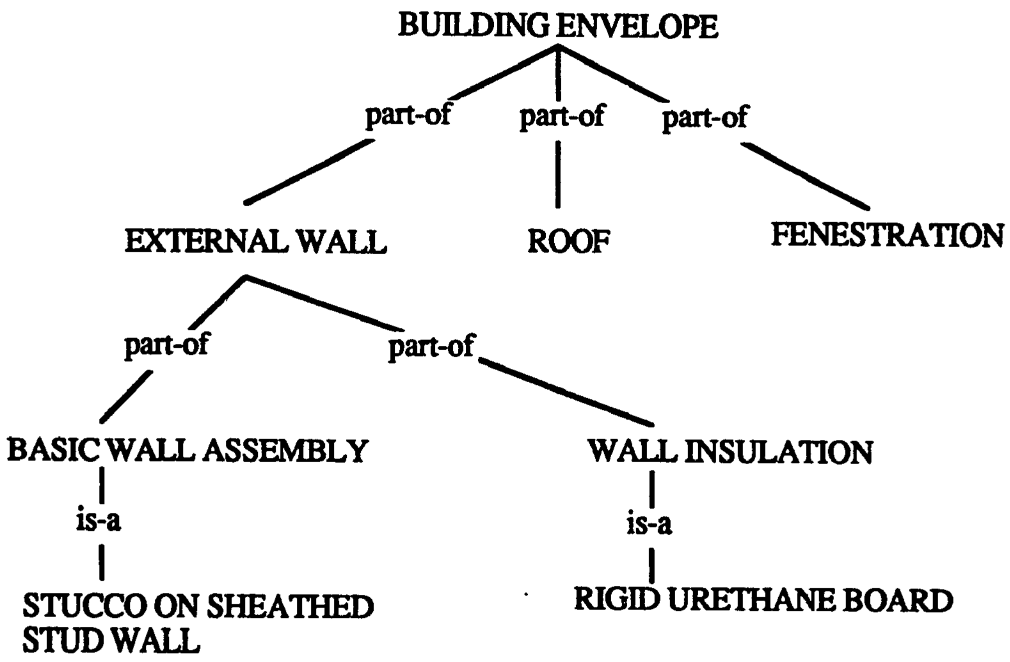


Figure 2.5: Hierarchy of Building Envelope Components

the process and the fact that designers prefer to have the controls of decision making. Design decisions are often influenced by the four factors [44]:

(i) Outside events and agencies, and other constraints.
(ii) Experience
(iii) Personal choice and tradition
(iv) Recorded design data

Hence any computerized design decision must allow the designer to express preference on materials and systems, and to assign priorities on performance attributes. This can be achieved by ranking the feasible alternatives according to each of the performance attribute considered and by applying a simple weighted utility criteria to account for designer priorities. Such a technique will be meaningful for comparing the alternatives on a relative performance basis rather than emphasizing the accuracy of predicted performance.

2.5.2 Scope and Limitations

The proposed approach aims at organizing the existing knowledge of building envelope design in order to create a design framework and a practical tool that can assist a designer in establishing the design context, defining the performance attributes, generating feasible design alternatives, ranking these alternatives and selecting the best suitable one, at the preliminary design stage.

The present work is aimed at developing a systematic approach to automating the design process, hence no attempt is made to develop or evaluate the available techniques for predicting a particular aspect of building envelopes performance. But some simple and commonly used performance evaluation techniques are implemented in the prototype system to demonstrate the applicability.

BEADS is a prototype implementation of the proposed methodology for the selection of materials and constructional types for external walls, roof and glazing systems of building envelopes. The knowledge base of the BEADS consists of various types of information representing the following:

(i) Design weather data
(ii) Energy efficient design requirements
(iii) Basic construction type descriptions
(iv) Material properties
(v) Design heuristics

The proposed system assumes that decisions on the size of building, occupancy, structural system and area of envelope components are already made. Further it assumes that the designer is interested in exploring design alternatives for the envelope assembly. A characteristic feature of the proposed methodology is due to the minimum amount of information required to define the design context which is often the case during the early stages of design, and the ability to progressively refine the performance objectives as the design proceeds. A design query language user-interface is developed for providing the flexibility to examine the influence of design parameters in the alternative generation, evaluation and selection processes.

The BEADS knowledge base is developed from design handbooks and by interviewing two practising designers. The present work considers only the design of major envelope components and hence the construction details, joints, sealant, and other aspects are not addressed. Only a few performance attributes are considered in the prototype for demonstrating the feasibility of the proposed approach. Though the knowledge base of the prototype system realistically represents all the information required for energy efficient building envelope design, it is certainly not complete and needs more knowledge acquisition to account for other performance attributes which are not considered presently.

CHAPTER 3

KNOWLEDGE-BASED SYSTEM DEVELOPMENT TOOLS

3.1 Introduction

The traditional approach to automate design processes with procedural programming languages results in algorithm intensive problem solving which provides a predetermined design solution using the strategies identified during the program development. Programs developed with this approach are of limited use in practice, and are not amenable for easy updating and maintenance. Recent advancements in software development and knowledge-based system (KBS) techniques provide the capabilities to overcome some of the difficulties in automating the design process. The potential of KBS techniques for solving engineering design problems is well explored and many prototype applications have been reported ([34], [35], [45], [46]). A number of software environments are commercially available to assist in KBS applications. Some of the currently available KBS development tools have been evaluated in the present study to select a suitable one for implementing the proposed building envelope analysis and design system. The characteristics of KBS development tools in general, the details of evaluation process and description on the features of the tools considered, are presented in the following sections.

3.2 Characteristics of KBS Development Tools

KBS architecture for engineering design applications is described in detail by Sriram et al [46]. The essential components of a KBS are the knowledge base, the inference mechanism and the user-interfaces. Earlier attempts to develop KBS applications were faced with the non-availability of suitable software tools other than the symbolic programming languages LISP and Prolog. Development of KBS applications using LISP and Prolog requires the developer to devise the knowledge representation and inferencing methodologies for the problem and the user interface. Hence special purpose development environments such as BUILD [36], were developed to satisfy the needs of a particular application domain. The required time, effort and programming expertise for developing such special purpose environments are unjustifiable and may not be successful. Presently, a number of general purpose KBS development tools are available. The capabilities of these tools vary widely depending on the knowledge representation and inferencing methodologies, user friendliness of the developer interface and the features for the development of end-user interface.

Many knowledge representation techniques for encoding domain specific knowledge have been developed. The two most widely known methodologies are based on production rules and frames. Most of the available development tools provide the capability for representing knowledge in the form of production rules. Production rules are useful in representing Code specifications and heuristics of design. Frame representation is suitable for specifying the properties of objects and in semantically relating the object for inheritance. For example, a frame "BUILDING-MATERIAL" may consist of slots corresponding to the properties of a building material, and these properties will be inherited by a frame "BRICK", which is an instance of building material. Thus the structure of a knowledge base can be developed using taxonomies for representing the information about the problem domain. Hybrid systems providing both production rules and frames are also available. The most widely used inference mechanisms are based on forward and backward chaining techniques. Forward chaining is useful for data driven problems and backward chaining is efficient for goal driven problems. A combination of both forward and backward chaining is also provided by some of the recent tools.

A knowledge base editor is required in the development environment for creating and modifying the knowledge base. Another desired feature in developing a successful application is the possibility to create a friendly end-user interface without much programming efforts. Typical options required for an end-user interface development are automatic input-validation, range and type checking, providing explanations for prompts and automatic reasoning capabilities. KBS development tools are primarily aimed at symbolic processing and many of them do not provide computational capabilities, interface to data bases or external programs which are often required in engineering design applications. The advantages and limitations of the available development tools are specific to the requirements of an application. Hence a number of prototype applications are developed in various development tools to identify the suitable one for use in the present study. Eight KBS development tools - OPS5, KNOWLEDGE CRAFT, GURU, PC PLUS, GOLDWORKS, M.1, INSIGHT 2+ and NEXPERT OBJECT - are considered and prototype applications in developing design assistants for reinforced concrete design, snow load estimation and material selection are developed. The features of each tool, details of prototype implementations and a comparison of the tools are given below.

3.3 Evaluation of Development Tools

3.3.1 OPS5

OPS5, which stands for Official Production System 5, is one of the most popular and successful knowledge-based system development tools. In the development of earlier applications such as HI-RISE [34], the rule base was developed in OPS5. Knowledge representation in OPS5 uses an object-attribute-value tuple. The inference mechanism is primarily based on forward chaining, with options to control the conflict resolution scheme. There are many versions of OPS5 available on a wide range of hardware. In the present study, three versions of OPS5 were investigated. OPS5, available in the VAX-VMS environment is a BLISS-32 implementation [47]. The interface to external functions or programs with this version was found to be extremely difficult and time consuming. The version available on the IBM personal computers, known as OPS5+ is implemented in C [48], and this version requires the user to be familiar with C in developing external functions or interfacing to other programs. ExperOPS5 is an ExperLISP implementation of OPS5 available on the Apple Macintosh computer [49]. A routine calculation step such as finding the square root of a number cannot be easily achieved with the BLISS-32 and C implementations of OPS5. But the ExperLISP implementation provides the facility to add LISP functions as part of the knowledge base. The library of LISP functions available under ExperLISP include the arithmetic and triganometric functions. This is essential for engineering design applications which involve numerical computations during the design process.

The knowledge base in OPS5 consists of production rules and is known as production memory. The problem data is represented by an element class with associated attribute-value pairs known as working memory. Details of OPS5 programming can be found in the book by Brownston et al [50]. The development features provided by OPS5 are similar to that of a typical programming language excepting that OPS5 is specifically aimed at production rule programming. OPS5 does not provide any facility for rule editing, syntax verification or debugging except for a 'watch' facility to keep track of the changes to the working memory. The generation of explanations or any facility required for end-user development have to be thought out in the design of the knowledge base. This means that the amount of programming effort required would be considerably high and would influence the flexibility of the knowledge base for future updating of the application.

In the present evaluation, a reinforced concrete design program is developed to study the feasibility of integrating Code specifications and computations required in detailing reinforced concrete beam sections. By specifying the cross section dimensions, the design moment, and the material properties etc., the user could obtain the reinforcement details. The Code specifications for minimum and maximum reinforcements permissible, the clear cover requirements for reinforcement, and the computations for obtaining the area of reinforcement, as specified by the clauses 10.1 through 10.6 of the Canadian Standard CAN3-A23.3-M84 [51], are encoded in the form of production rules and ExperLISP functions. The knowledge base for simple beam design consisted of twenty production rules and five LISP functions. Example of an OPS5 rule, corresponding to clause 10.3.3, is shown in Figure 3.1. This rule checks the required percentage of steel reinforcement against the maximum permitted by the Code and warns the user. A goal oriented MEA (means-ends-analysis) strategy was used in implementing this program. Even though the reinforced concrete beam design is a simple problem, it involves a considerable amount of Code checking and for satisfying the various constraints related to bar diameters, bar spacing and constructability requirements. The knowledge-based system approach with OPS5 provides the facility to incorporate the Code specifications and design computations in a fashion that each can be treated as independent in terms of production rules and LISP functions for development and updating, but integrated together for designing the beam section. In the present implementation, an attribute "status" was assigned to the design variables, hence the design revisions could be done by modifying the value for this "status". Such facilities can provide the designer, a complete control of the design process, at the same time automating the Code specifications and computations. The development of an application using OPS5 requires a complete understanding of the limitations of the systems flexibility in knowledge representation and end-user interface development capabilities. Interfacing to programs written in other programming languages and generation of

```
(p  rqd-rho-to-rho-max-check
    (input ^name rho-max ^status known ^value <x>)
    {<goal> (input ^name rqd-rho ^status known ^value {> <x>}) }
-->
    (write (crlf)) (write (crlf) "*** WARNING ***")
    (write (crlf) "Rqd % area of steel is greater than allowable")
    (write (crlf) "maximum. Refer to cl.10.3.3 of CAN3-A23.3-M84")
    (modify <goal> ^status active ^value 0))
```

Figure 3.1: Example of a OPS5 Rule

explanations were not investigated in this study, but must be considered in the development, if these problems are likely to be encountered.

3.3.2 KNOWLEDGE CRAFT

KNOWLEDGE CRAFT [52] is a frame-based development tool integrating schema representation language with rule-based, logic and object-oriented programming methodologies. This tool enables the development of large scale KBS applications by allowing flexible knowledge representations, problem solving techniques and control strategies. KNOWLEDGE CRAFT is available on mini computers and micro workstations. In the present study, a VAX/VMS version of KNOWLEDGE CRAFT available in CommonLISP environment is used. KNOWLEDGE CRAFT consists of CRL, CRL-OPS and CRL-PROLOG. CRL is a schema representation language for frame-based knowledge representation and object-oriented programming. CRL provides the functions for creating and editing frames which can consist of a set of slots, and values corresponding to each slot. Procedural attachments in the form of LISP functions can be used as demons when obtaining a value for a slot. Relational links such as IS-A and INSTANCE are provided in CRL to enable inheritance between objects. The inheritance semantics may be customized by the user for special relations. Thus knowledge representation using frames can reduce the problems encountered in rule-based systems for semantically relating objects. Other functions in CRL include meta knowledge representation, context definitions, and agenda control mechanisms. CRL-OPS is forward chaining system, which is a superset of OPS5. The capabilities of CRL-OPS include that of OPS5 described in the earlier section. In addition, the production rules can act directly on the objects defined using CRL, and may interface with CRL-PROLOG which is a Prolog like backward chaining, goal driven system. Using CRL-OPS and CRL-PROLOG together, black board architectures for design integrating data driven and goal driven strategies can be implemented.

A prototype application for insulation material selection was implemented using KNOWLEDGE CRAFT. The user interface capabilities provided by KNOWLEDGE CRAFT to assist in development are complex and difficult for an inexperienced user. Hence a front-end devised for knowledge acquisition, known as ESCHER [53] is used in developing the knowledge base. ESCHER assists the developer using a menu driven strategy for user input and interface with CRL for creating and editing the frames, slots and values. A frame "INSULATION-MATERIAL" is defined for specifying the properties of insulation materials. This frame consists of six slots, each corresponding to a performance attribute and an additional slot specifying the suitablity of insulation for a particular type of construction. Figure 3.2 shows an example of "EXPANDED-POLYSTYRENE-BOARD", which is an INSTANCE of "INSULATION-MATERIAL".

Information about the properties of eight insulation materials from the Masonry Council of Canada's handbook [14] are specified in the knowledge base. Each property has a value that ranks the performance and suitability of a given material. A LISP function is developed to perform the inferencing and to control the end-user interaction. The user can specify the performance requirements of an insulation material for a particular situation, and this prototype will suggest a suitable insulation.

The use of KNOWLEDGE CRAFT for this problem is trivial but provides the information on the type of problems that could be implemented. Addition of more materials, their properties and a more systematic evaluation procedure can be incrementally developed without duplicating or redoing the previous work, saving the efforts required in updating and maintenance.

3.3.3 GURU

GURU is a knowledge-based system development tool developed for the business application software market [54]. GURU provides a development environment which integrates the capabilities of a rule-based system, natural language interpreter, data base management system, spread sheet analysis, business graphics, text processing, and procedural language programming facility. Knowledge base development in GURU proceeds in the following sequence: definition of a goal, an initialization phase, definition of program and environment variables, rules for the system and a completion phase. A menu interface with templates is provided to assist the developer-user in creating the rule base, data base, variable definitions, etc. Alternatively, a standard text editor may be used for development. The initialization phase is optional and it may contain instructions written in GURU command language and the programming language constructs. Usually, the introduction screen for an application and the variable initialization for the rule base are performed in the initialization phase. The contents of the initialization phase are executed before operating on the rule base. Rules in GURU

```
{{EXPANDED-POLYSTYRENE-BOARD
          THERMAL-RESISTANCE
                             LOW
          THERMAL-EXPANSION
                             HIGH
          COST
                             HIGH
          STRENGTH-RIGIDITY
                             LOW
          POROSITY-PERMEABILITY
                             VERY-HIGH
          FLAMMABILITY
                        HIGH
          SUITABLE-STRUCTURE-TYPE
                             CAVITY-WALLS
          IS-A
                             INSULATION-MATERIAL}}
```

Figure 3.2: Example of a Frame and its Slots in CRL

consist of an antecedent, consequent, reasoning and explanation description, and a list of variables that are required and/or modified by the rule. The antecedents and consequents may contain programming and command language instructions. For example, WHILE loops, IF clauses, reference to data base and spread sheet fields may all be used as part of an antecedent or consequent. This flexibility provides the capability to integrate data base information easily with the rule base, and to a certain extent permits procedural programming as part of a production rule. There are many environment variables one could use to control the inferencing and conflict resolution methodology. The definition of variables can be provided with a translation property and a set of instructions to be executed in determining the variable. This facility is useful for programming the input prompts and validations. The completion phase consists of instructions to be executed before completing a consultation with the knowledge base. The initialization and completion phase are complementing the rule base by grouping the work that needs to be done before and after reaching the goal defined in the rule base.The advantages of GURU is the flexibility due to integration and the versatility of its components.

In the present investigation, a snow load estimation system was implemented in GURU. The Part 4 of the National Building Code of Canada [55] specifies the procedures to determine the various types of structural loads acting on a building during its life span. Subsection 4.1.7 of the Code consists of details for estimating the intensity of load due to snow, ice and rain. The design snow load on a roof area is obtained by multiplying the ground snow load by a series of adjustment factors which depend on the wind exposure, roof type, roof slope and the accumulation characteristics. The Supplement to the National Building Code of Canada [56] provides the climatic information on the intensity of ground snow loads for about 480 locations across Canada. These ground snow load values have a thirty year return period and are based on observations and data collected at various stations across the country. The data management scheme in GURU is used to create a data base of city names and corresponding ground snow loads as specified in the Supplement. The data for 120 locations in the province of Quebec are currently stored in a table form. During the initialization phase of a consultation, the user will be asked to input the city name where the building will be located, and the ground snow load corresponding to that location will be retrieved from the data table. If the city name input by the user is not found in the data table, then a warning message will be given and the user may provide a valid city name closer to the location being considered. After obtaining the ground snow load, which is the basic data needed in this system, the control is transferred to the rule for obtaining the adjustment factors. The adjustment factors include the slope factor and accumulation factor which are based on the roof type and roof slope. The Code specifications for evaluating these adjustment factors are translated in the form of GURU rules. The user is provided with a list of six different roof types that are commonly encountered and discussed by the Supplement. More than one load case needs to be considered in the design of certain roof types such as arches. The amount of input information required to describe the roof geometry varies depending on the roof type. The variable definition in GURU provides a 'find' option which can be used to program the user prompt and input validation, when the value of a variable is not known. This reduces the number of rules required in developing a good end-user interface.

About 40 rules have been used in translating ten sentences, eleven clauses and three subclauses of the Code and in incorporating the details provided by the Supplement for snow load estimation. GURU provides a number of environment variables for controlling the inference methodology and a data driven, forward chaining is used in the snow load estimation system. A sample consultation session for estimating the snow load on a arched/curved roof is shown in Figure 3.3. The number of load cases and the details of loading are obtained as output, based on the location of the building and roof geometry. Arched/curved roof is one of the most difficult types for which the snow load estimation requires a careful interpretation of the Code. The GURU explanation facility, by default provides the values of variables for each of the rule used in the consultation process. Reasoning for each rule can be specified, but was not attempted in this study.

The advantage of GURU for the snow load estimation problem is due to the facility for integrating the data table and the rule base with minimum programming effort. Use of a data table for climatic information makes book keeping easier. This data table can be extended to incorporate other design information such as design temperatures, wind pressure, etc., and this data table can be used by more than one rule base. Significant savings in data base creation and maintenance by avoiding the duplication of information are immediate advantages of this approach. Whenever the Code is revised or updated with changes to the climatic data, the corresponding changes to the data table can be done without modifying the rule base.

```
          Welcome to the snow load estimation routine

Where is the building located?     Baie-Comeau

Choose the type of roof from: 1. Flat roof
                              2. Shed/single or sloped roof
                              3. Gable roof
                              4. Arch roof
                              5. Valleys in roof
                              6. Multi-level roof

Input the type of roof:     4

What is the height of arch(m)?     4.0

What is the span of the arch(m)?     18.0

What is the edge slope of the arch (in deg.)?     45.0

What is the h30 for this arch?     2.5

Load Case 1:
      Snow load at mid span .........     4.1 kPa
      Snow load along the edges .....     2.5 kPa

Load Case 2:
      Snow load (maxm. accumulation).     6.00 kPa
      Snow load along the edge ......     3.75 kPa

(Refer to Fig.H-2 of the NBCC Supplement)

*** End of snow load computation ***
```

Figure 3.3: Example Consultation for an Arch Roof

3.3.4 PC PLUS

Personal Consultant Plus (PC PLUS) is a knowledge-based system development tool available from Texas Instruments Inc.[57]. PC PLUS is a rule-based system development tool with an interactive and menu-driven interface for the developer and the end-user. The knowledge-based system components of PC PLUS are provided in a LISP environment known as PC SCHEME. Development of a knowledge base for an application using PC PLUS requires the definition of environment variables, their properties, goal parameters and the rules required to solve the problem. The developer is prompted for providing all the information required to develop an application. For example, the definition of a variable will be initiated by the system, when it encounters an undefined variable name in a new rule that is added to the knowledge base. There are a number of properties which can be assigned to a variable. Some of the most useful ones are the natural language translation property, list of legal values and consultation prompt. Rules in PC PLUS have antecedents and consequents written in ARL (Abbreviated Rule Language) form or LISP form. Figure 3.4 shows an example of a rule in natural language, ARL and LISP forms. The ARL form is used by the menu interface in developing the rule base, and this will be translated by the system to produce the LISP form which is executed in the PC SCHEME environment. The natural language form is generated using the translation property of the variables used in a rule.

LISP expressions may also be included as part of the antecedent or consequent of a rule. This facilitates computational and iterative programming required in engineering design, but assumes a familiarity with PC SCHEME. Rules in PC PLUS may be grouped to form a rule set and meta-rules can be used to control the use of rule sets. Every rule set will act as an independent system, but the variables and their values can be shared between rule sets. Hence the integration of future developments can be planned without much difficulty. PC PLUS provides a DOS call option with which operating system commands may be executed. This facilitates a possible communication link using a disk file for data transfer between an external program and a knowledge base. PC PLUS provides an interface for DBASE files and for incorporating graphic information in the knowledge base.

The debugging of a knowledge base created in PC PLUS is facilitated by the REVIEW, TRACE and PLAYBACK options. The system can generate reasoning and explanation for WHY and HOW options, using the natural language translation property

NATURAL LANGUAGE FORM:

```
RULE005 [RC-WALL-RULES]
-------
If unsupported width of the wall is less than the unsupported
height of the wall,
Then it is definite (100%) that thickness of wall in mm is
[unsupported width of the wall divided by 25.0].
```

ARL FORM:

```
IF: VALUE W < VALUE H
THEN: TW = VALUE W / 25.0
```

LISP FORM:

```
PREMISE:    ($AND
              (LESSP*
                (VAL1 FRAME W)
                (VAL1 FRAME H)))
ACTION:     (DO-ALL
              (CONCLUDE FRAME TW
                (FQUOTIENT
                  (VAL1 FRAME W) 25.0) TALLY 100))
```

Figure 3.4: Example of a Rule in PC PLUS

of the variables used. The conclusion screen provided by PC PLUS is based on the goal parameters. More than one goal parameter can be defined and the system tries to resolve all the goal parameters during a consultation. The inferencing methodology used in PC PLUS is primarily backward chaining with an option to define antecedent rules. Because of the effective user interface capabilities, application development using PC PLUS will be easier and faster than other rule-based systems.

A reinforced concrete wall design system was developed using PC PLUS. For axially loaded walls, an empirical design method for determining the wall thickness and calculating the reinforcement details is specified by clause 14 of the reinforced concrete design Code, Canadian standard CAN3-A23.3-M84 [51]. As the design method is empirical, this is an ideal case for representing Code specifications and the design criteria in the form of production rules. An example showing the input parameters list and the output for the consultation are shown in Figure 3.5. The input required for the design consists of the wall geometry, support conditions and applied load. The overall thickness of the wall and the load resistance of the wall can be calculated and verified for using the empirical method. Then the user may specify the required bar size and obtain the reinforcement details for the wall. About thirty rules and twenty five parameters are used in developing the reinforced concrete wall design system. The REVIEW option can be used for modifying any of tne already input parameters and the consultation can be carried out. This is very useful in reinforced concrete design where the designer may decide to change the grade of concrete or the bar size in revising a design. Any additional information required during the design revision will be inferred by the system. Hence the effort required in revising a design is minimized by reviewing a previous design and specifying the desired modifications.

3.3.5 GOLDWORKS

GOLDWORKS is a knowledge-based system development tool with a hybrid knowledge representation methodology using frames and rules. Very few tools available for the personal computers provide this hybrid knowledge representation capability. The application development features of GOLDWORKS are provided in a LISP environment known as GCLISP [58]. GOLDWORKS has two levels of development user interface: non-programmers are provided with a menu driven interface to create frames, rules and objects for the knowledge base, and users familiar with LISP can use a top-level

INPUT FOR A SAMPLE PROBLEM:

```
("Consultation record for: Reinforced concrete design of walls"
"compressive strength of concrete          ::    30"
"unsupported height of the wall            ::    3000"
"unsupported width of the wall             ::    5000"
"wall is cast on site                      ::    YES"
"centre to centre between point loads...   ::    2000"
"bearing width in mm                       ::    600"
"top edge restrained against lateral ...   ::    YES"
"bottom edge restrained against later...   ::    YES"
"top edge restrained against rotation      ::    YES"
"applied load in kN                        ::    300"
"Preferred bar No. for reinforcement       ::    20"
```

CONCLUSION SCREEN:

```
RC-WALL-1  CONCLUSIONS:
Thickness of wall in mm is as follows:   150

Design width in mm is as follows:   1200

Factored axial force resistance in N is as follows:   2004750

Area of steel rqd. for horizontal direction in mm is as follows:300.

Area of steel rqd. for vertical direction in mm is as follows: 225.

Preferred bar No. for reinforcement is as follows:   20

Horizontal reinforcement spacing is as follows:   450

Vertical reinforcement spacing is as follows:   450

Use of empirical design method is as follows:   SUCCESSFUL
```

Figure 3.5: Example of Input and Conclusion for RC Wall Design

developer interface which provides a number of built-in functions and a GMACS editor for development. The knowledge base developed using the menu interface is translated to LISP, and the execution is performed by GCLISP. The development of an application using GOLDWORKS requires the definition of objects, frames, instances and rules. GOLDWORKS treats frames consisting of slots, as templates. The slots have facets which are used for defining the default value for the slot, legal values that can be assigned to it, and procedural attachments as demons. Instances of a frame is created to provide the knowledge about a physical entity in the problem world, by specifying values for the slots. Rules in GOLDWORKS operate on the instances of frames. Rules may contain LISP expressions which are evaluated during the program execution. The inferencing methodology can be specified for an application and it may be forward chaining, backward chaining or an integrated forward/backward chaining. Assertions, agenda items, attempts, sponsors and relations are some additional features available for controlling the inference mechanism and structuring the knowledge base. A screen tool kit of predefined frames is available for end-user interface development, with which pop-up menus, command lines, multiple windows, and scrollable screens can be created. GOLDWORKS provides the capabilities to interface with DBASE and LOTUS-123 data files and to interface with functions in C. Any DOS operation can be performed while in the GOLDWORKS environment.

For evaluation purposes, an insulation material selection adviser as described earlier in section 3.4 (KNOWLEDGE CRAFT) was implemented in GOLDWORKS. Figure 3.6 shows the GCLISP function generated by the developer interface in defining the frame "BUILDING-MATERIAL", and "INSULATION" which is a building material. Instances of this frame were used to represent the properties of various commercially available insulation materials. User input for this system is through an interactive session which prompts the end-user and provides a menu type input facility, generated using the screen tool kit. Input validations are automatically done as the values of the slots are constrained by those specified in the "constraints" facet.

GOLDWORKS inferencing methodology is built in such a fashion that any change to the assertions will carry out the inference process automatically, to satisfy the new situation. For example, assume that a user specified a set of performance levels and obtained an insulation material suggested by the system. If the user now modifies any one of the six performance attributes, the system will automatically update the conclusion by providing the name of a material which successfully meets the new set of

```
(DEFINE-FRAME BUILDING-MATERIAL
  (:print-name "BUILDING-MATERIAL"
   :doc-string ""
   :is TOP-FRAME)
  (FLAMMABILITY
    :default-values    (UNKNOWN)
    :constraints (:ONE-OF (VERY-HIGH HIGH MEDIUM LOW VERY-LOW UNKNOWN)))
  (POROSITY-PERMEABILITY
    :default-values    (UNKNOWN)
    :constraints (:ONE-OF (VERY-HIGH HIGH MEDIUM LOW VERY-LOW UNKNOWN)))
  (STRENGTH
    :default-values    (UNKNOWN)
    :constraints (:ONE-OF (VERY-HIGH HIGH MEDIUM LOW VERY-LOW UNKNOWN)))
  (COST
    :default-values    (UNKNOWN)
    :constraints (:ONE-OF (VERY-HIGH HIGH MEDIUM LOW VERY-LOW UNKNOWN)))
  (THERMAL-EXPANSION
    :default-values    (UNKNOWN)
    :constraints (:ONE-OF (VERY-HIGH HIGH MEDIUM LOW VERY-LOW UNKNOWN)))
  (THERMAL-RESISTANCE
    :default-values    (UNKNOWN)
    :constraints (:ONE-OF (VERY-HIGH HIGH MEDIUM LOW VERY-LOW UNKNOWN))))

 (DEFINE-FRMAE INSULATION
    (:print-name "INSULATION"
    :doc-string ""
    :is BUILDING-MATERIAL))
```

Figure 3.6: Example of a Frame Definition in GOLDWORKS

requirements. The modification to the value of a slot is simple, but must be anticipated by the developer in the design of the end-user interface.

3.3.6 Other Tools

Three other tools - M.1, INSIGHT 2+ and NEXPERT OBJECT - were also investigated in this study. An initial evaluation and experience with the tutorial provided by these tools showed severe limitations for engineering design problems. Hence no attempt was made to develop any complete prototype application with these tools. A brief description of the characteristics of each of these tools and their limitations are given below.

M.1[59] was one of the first development tool available for personal computers. This is a rule-based system tool with a backward chaining methodology. A knowledge base in M.1 consists of facts and rules. There are a few options such as 'legal values' and 'question' for variables which need to be input by the end-user. As M.1 does not provide a knowledge base editor, the use of a standard text editor is mandatory to create and edit a knowledge base. The debugging of a knowledge base in M.1 is enhanced by the 'trace' facility which provides the details of an execution in a multiple window display. Computational capabilities of M.1 are limited to simple arithmetic operations. The snow load estimation system described earlier was first attempted in M.1 and abandoned because of the large number of facts required to incorporate the ground snow load information. M.1 provides an option to interface with C and assembly language routines, but the process is difficult and time consuming.

INSIGHT 2+ [60] is another rule-based system development tool with backward chaining methodology. This tool provides more facilities to the developer than M.1 in terms of knowledge base editing, compilation of the knowledge base and interface to procedural language programs. Rules in INSIGHT 2+ could contain an ELSE clause, and facts may be represented as an object-attribute type. Though INSIGHT 2+ provides the necessary features for developer interface, its use for automating Code specifications is limited on the basis of restricted knowledge representation and inferencing methodologies which are in fact similar to that of M.1.

NEXPERT OBJECT [61] is a hybrid system providing frames and rules for knowledge representation. A menu driven developer interface is provided by this tool in the Microsoft Windows environment. A rule editor, context editor, object editor, class editor and property editor are available for creating the knowledge base with an option to graphically display the objects and rules in a network form. NEXPERT OBJECT has a number of additional features such as interface to data bases and procedural language forms, communication with other processes, and for customizing the interfaces. As the system is completely menu driven, and a number of windows are simultaneously opened and displayed, it is difficult to become familiar with. In fact, the development environment provides so many options to be user friendly that it appears counter productive.

3.4 Comparison of the Development Tools

A qualitative comparison of the tools evaluated in this study is presented in Table 3.1. The basis for comparison is the experience gained in developing prototype applications. All the development tools except KNOWLEDGE CRAFT, reported in this study can be used on an IBM PC and compatibles with a hard disk for secondary storage. Any specific hardware and software requirements for the development tools are given in the "Remarks" column of Table 3.1. The most important criteria to be considered in the selection of a development tool are the knowledge representation and inferencing methodologies. Other features such as computational capabilities, user interfaces, etc., can also become essential, depending on the requirements of a specific application. Any of the available development tools could be used for design applications, but some would be expensive, while others would require excessive programming effort and development time with severe restrictions on the performance and sophistication of the end product. In general, knowledge-based system developers are forced to tailor the application to fit into the capabilities of the tool that is used for development. This is a serious pitfall for engineering design developments and may result in a waste of time and efforts. This is why it was considered necessary in this study to implement prototype applications to evaluate each of the development tools for engineering design.

OPS5 provides limited facilities for performing computations and interfacing to external programs. A standard text editor, outside the OPS5 environment, has to be used for creating and updating the knowledge base. Even though OPS5 provides a more

Name of Development tool	Knowledge representation methodology	Inference mechanism	Computational capabilities	Data base interface	External program interface	Developer interface & debugging facilities	End-user interface development capabilities	REMARKS
OPS5	Production rules	Forward chaining	Simple arithmetic	Not available	Not available	Trace available for debugging	Not available	Many of the features are dependent on the particular implementation of OPS5
KNOWLEDGE CRAFT	Frames and Production rules	Facilities of OPS5 and PROLOG	CommonLISP functions	Avalaiable	Available	Trace and functions available	Not available	Available in mini and workstation computers only
GURU	Production rules	Forward chaining, backward chaining & a combination	Full set of programming features	Integrated	Not available	Command and menu interface, and trace available	Not available	—
PC PLUS	Production rules	Backward chaining with option for antecedent rules	PC SCHEME functions	DBASE	DOS-CALL	Menu interface, trace review and playback	Available	—
GOLDWORKS	Frames and production rules	Forward chaining, backward chaining & a combination	GCLISP functions	DBASE and Lotus 123	DOS-CALL and C/PASCAL	Menu and LISP level developer	Available	5 MB min. RAM and 10 MB min. disk spac
M.1	Production rules	Backward chaining	Simple arithmetic	Not available	C/Assembler	Trace available for debugging	Not available	—
INSIGHT 2+	Production rules	Backward chaining	Simple arithmetic	Not available	PASCAL	Menu interface and trace available	Not available	—
NEXPERT OBJECT	Frames and production rules	Integrated forward and backward chaining	Simple arithmetic	Customized data base interface	C	Menu/window interface	Not available	MS WINDOWS program required

NOTE: Uncertain information handling, explanation and reasoning facilities have not been considered in this investigation.

Table 3.1: Comparison of knowledge-based system development tools for design applications

powerful knowledge representation methodology than other rule-based system development tools, it may not be the best choice for use in engineering design because of the rigid development environment. OPS5 is used for learning expert systems programming and in AI research [50]. KNOWLEDGE CRAFT provides the capabilities for large scale developments with options for versatile knowledge representation and inferencing techniques by integrating CRL, CRL-OPS and CRL-PROLOG. Interfacing to external programs, data bases, and engineering computations can be performed using CommonLISP, in which the KNOWLEDGE CRAFT is implemented. But the development environment does not favour a new user and requires LISP programming knowledge. These requirements can be overcome with a user interface such as ESCHER, for rapid prototyping and development of applications using KNOWLEDGE CRAFT. GURU provides most of the capabilities required for developing rule-based system applications that need to integrate with data bases. The disadvantages of GURU are mainly the absence of external program interfaces and the extensive programming effort required to develop a good end-user interface. Because of the large number of options provided, beginners may find the time required for learning GURU to be excessive. PC PLUS is one of the most user friendly development tools evaluated in this study. If simple rule-based representation is sufficient for an application, then PC PLUS is an ideal choice. However, PC PLUS is found to have sluggish I/O operations when the number of rules increases, thus making it not advisable for large scale developments. Also knowledge of PC SCHEME - LISP is required to perform iterative computations and to use any of the advanced features such as interfacing to data bases and external programs. GOLDWORKS provides a versatile knowledge representation methodology using frames and rules. Multiple levels of user interface capabilities are available depending on the user's familiarity with the environment and this becomes useful for rapid developments. Computational and external program interface capabilities can be achieved using GCLISP, which therefore has to be learned by the user to take advantage of the advanced development features. If the extensive hardware and software requirements of GOLDWORKS can be met, then this becomes a suitable tool for the development of engineering design applications.

M.1 and INSIGHT 2+ have limitations that are inherent to simple rule-based system development tools, which are suitable for diagnostic-type problem solving because of the restricted inferencing methodology. These tools also have limited capabilities for developing user interfaces, though INSIGHT 2+ could be considered to have more features than M.1. NEXPERT OBJECT provides a number of useful capabilities as a hybrid system development tool, but was found difficult to use because of a complex user interface.

3.5 Selection of a Development Tool

The choice of a specific tool must be based on the requirements of a given application. Rule-based systems in general are sufficient for Code compliance checking problems. However, the complexity of rule-based systems increases with the number of rules and may become unmanagable for large applications [62]. A more versatile knowledge representation methodology using multiple paradigms is desirable for general design type applications [63], which include Code compliance checking as a component. Based on the experience gained in developing prototype applications, KNOWLEDGE CRAFT is chosen for implementing the proposed building envelope design system. The hybrid knowledge representation facility and the potential for long term development are the major advantages of using KNOWLEDGE CRAFT. Many of the other tools, except GOLDWORKS lack the knowledge representation features. Though GOLDWORKS provides most of the desired development characteristics, it is expected that the speed of execution and development will be much lower than what could be achieved with KNOWLEDGE CRAFT.

CHAPTER 4

KNOWLEDGE REPRESENTATION

4.1 Introduction

Building envelope design process requires many different types of information for establishing the design context and in generating the design alternatives. This chapter presents in detail the design parameters and performance attributes considered in the BEADS prototype development. A schema/frame-based technique is employed to represent the knowledge about envelope components, materials, their properties, compatibility and semantic relationship between them. The details of knowledge acquisition and representation for the various stages of design in establishing the design context, generation, ranking and selection of alternatives are described in the following sections.

4.2 Design Context

During the preliminary design stage, very little information regarding the building such as its location, geometry, type of occupancy and budget are available. The definition of quantitative performance requirements at this stage depends on the experience and knowledge of the designer to make appropriate assumptions. Design handbooks and performance standards are available to assist the designer in this process of establishing the design context.

4.2.1 User requirements

It is necessary first to identify the parameters which are available and required in design. The following list of user requirements is usually known during the early design stages:

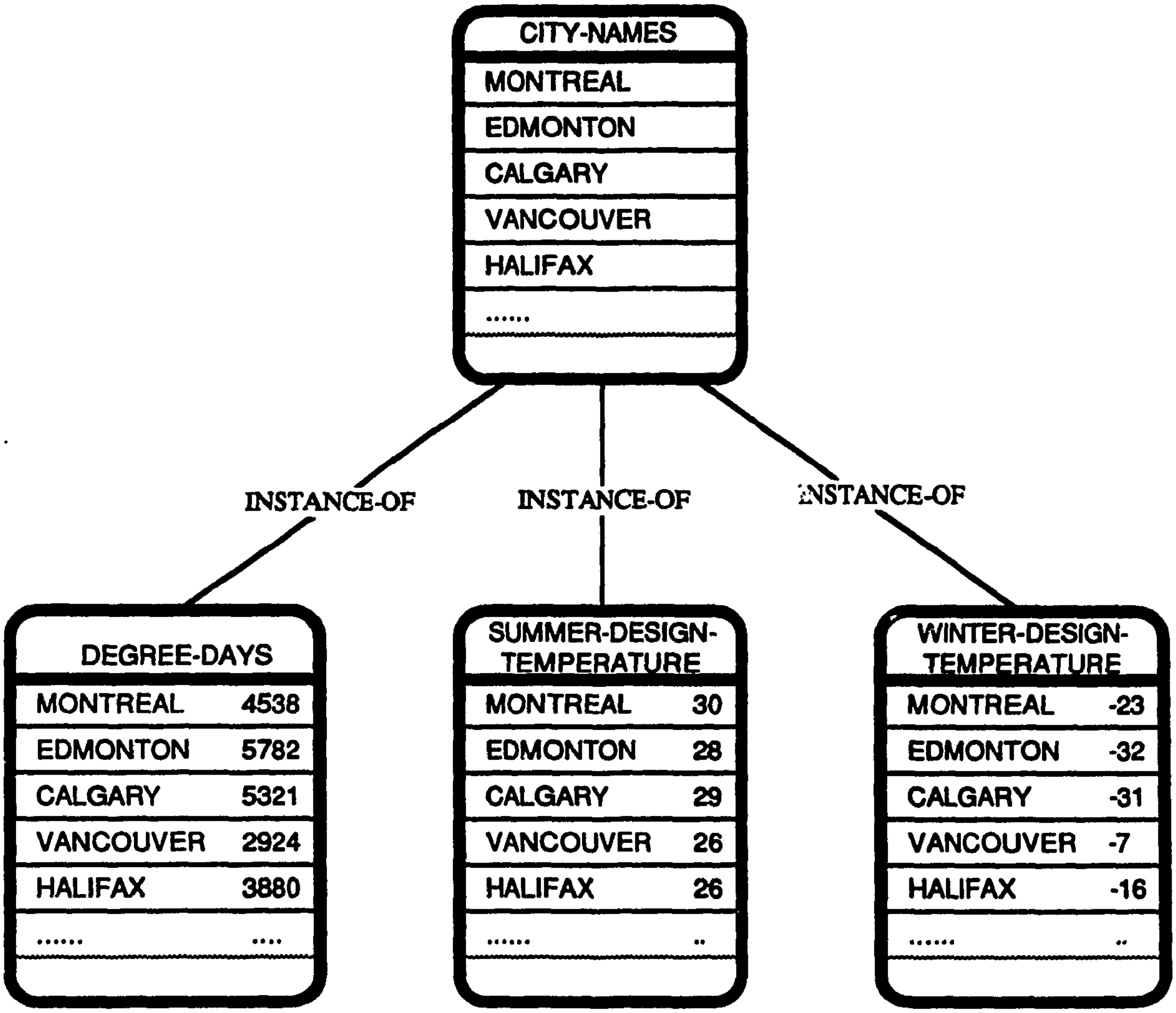
- Building location

- Building type and occupancy
- Structure type
- Roof slope/shape
- Gross area of external wall
- Gross area of roof
- Gross area of fenestration
- Permissible thickness of wall
- Permissible thickness of roof

The BEADS prototype system is developed assuming that only the above parameters are available for design. The quantitative performance requirements to meet the various functions of the building envelope can be derived from the above parameters and using experiential knowledge, Codes and Standards. The present prototype considers performance attributes relating to thermal resistance, moisture protection, cost, material compatibilities and thickness requirements. Some of these requirements can be established as target values in the design context whereas others can be determined only during the design alternative generation process. For example, the check for interstitial condensation and material compatibilities can be performed during the alternative generation process, hence they may be represented as procedures and part of the material properties respectively.

4.2.2 Weather Data

Design weather data including degree days, summer and winter design temperatures are required for the energy budget and condensation check calculations. The supplement to the National Building Code of Canada [55] provides data tables from which design weather data can be retrieved simply by knowing the building location. A schema "CITY-NAMES" is defined to consist of slots corresponding to city names. Instances of this schema are used to represent the various design weather data items such as design degree days and design temperatures. Figure 4.1 shows a graphical representation of the weather data as currently available in the knowledge base. The three schemas - "DEGREE-DAYS", "WINTER-DESIGN-TEMPERATURE" and "SUMMER-DESIGN-TEMPERATURE"- are defined with a relational link to the "CITY-NAMES" schema, thereby inheriting the location names. Then the data corresponding to each location and design parameter can be specified as the value of the attribute in the respective schema.



CITY-NAMES
MONTREAL
EDMONTON
CALGARY
VANCOUVER
HALIFAX
......

DEGREE-DAYS	
MONTREAL	4538
EDMONTON	5782
CALGARY	5321
VANCOUVER	2924
HALIFAX	3880
......	....

SUMMER-DESIGN-TEMPERATURE	
MONTREAL	30
EDMONTON	28
CALGARY	29
VANCOUVER	26
HALIFAX	26
......	..

WINTER-DESIGN-TEMPERATURE	
MONTREAL	-23
EDMONTON	-32
CALGARY	-31
VANCOUVER	-7
HALIFAX	-16
......	..

Figure 4.1: Weather Data Representation

Knowledge representation of weather data using the above methodology offers many advantages for incremental development, easy updating and retrieval. There are about 80 major locations in Canada currently available in the prototype system. More locations can be added to the "CITY-NAMES" schema by creating new slots which will be inherited by the related schemas representing the weather data. The addition of more environmental data specific to a location such as wind pressures and ground snow load can be done by defining new schemas and relating them to the "CITY-NAMES" schema. Then the corresponding values can be specified for each location.

During the initial stages of BEADS development, it was sufficient to have the degree days and temperatures information for establishing the design context. But later in refining the preliminary prototype it has been found that a cooling check needs to be performed for high rise buildings. In order to perform the cooling check according to ASHRAE 90A-1980 [43], the latitude of location is necessary. Then a schema "DESIGN-LATITUDE" is defined and latitude for the various locations are specified. Thus extending weather data information can be achieved with very little effort. If the designer specifies the location of building, then all the weather data required for establishing the design context can be retrieved. This approach reduces the amount of user input and at the same time offers the facility to examine the influence of location on the outcome of design.

4.2.3 Energy efficiency requirements

Building envelope (icluding windows) contributes significantly to the amount of energy consumed by the space heating and cooling systems ([13], [14]). The need for energy efficient design of envelope components has long been recognized and the regulatory authorities have developed standards for the optimum use of energy. ASHRAE Standard 90A-1980 [43] for energy conservation in the design of new buildings provides the minimum thermal performance requirements for the envelope components. The intent of this standard is to be flexible in order that designers be encouraged to use innovative approaches and techiques to achieve effective utilization of energy. Though the requirements specified by this ASHRAE standard are not aimed at optimized performance for energy conservation, they provide a basis for considering energy efficiency at the preliminary design stage. This standard covers new buildings that provide facilities of shelter for public assembly, educational, business, mercantile, institutional and residential occupancies as well as portions of warehouse, factory and industrial occupancy which are primarily used for human occupancy. Buildings are classified into the following four broad categories:

- (i) Detached one or two family dwellings
- (ii) Residential buildings - three stories or less
- (iii) Other buildings - three stories or less
- (iv) Other buildings - more than three stories

Guidelines and graphs are provided by this standard to determine the overall thermal transmittance (U_0 values) for the envelope components. The type of building and degree days of location can be used to find the U_0 values for gross area of walls and roof. Many preliminary design aids ([64], [65]) have been developed based on this ASHRAE standard [43]. Empirical relation such as the following equation can be used to meet the overall heat loss requirements ([64], [65]).

$$U_o^* = \frac{U_{wall} * A_{wall} + U_{fenestration} * A_{fenestration} + U_{roof} * A_{roof}}{A_{wall} + A_{fenestration} + A_{roof}}$$

The U_0 values of walls and roof provided by the standard are meant for the gross area of envelope, whereas it is seldom possible to meet this U_0 requirement for windows, doors and skylights. Hence area averaging technique using the above equation is ideal for examining different combinations of envelope components. For example, the individual U_0 requirements for wall and roof can be relatively increased or lowered beyond the minimum requirements specified by the standard, but the combination must satisfy the overall U_o^* requirement.

The wall and roof U_0 values, the overall thermal transmittance value (OTTV) and solar factor are obtained from the ASHRAE standard and defined in the design context as performance requirements. The U_0 values and the gross area of envelope components are then used to establish an energy budget. During the alternative generation process, the heat loss for each combination of envelope components will be calculated and checked against this energy budget. In addition, multistorey buildings belonging to category (iv) will be checked for cooling requirements using OTTV-wall values as specified in clause 4.4.3 of the ASHRAE standard [43].

The ASHRAE graphs to determine the above values can be represented in many different ways. These graphs typically require the building type and degree days of the location to determine the overall thermal transmittance values. Production rules or procedures can be used for representing the ASHRAE graphs. The present implementation uses LISP functions and the graphs are represented in procedural form. Figure 4.2 shows one such function for finding the Uo value of external walls of building type A1 (category 1). This function is accessed by the design context after knowing the user input on building type and location, and returns the appropriate Uo value depending on the degree days. Though production rules can also be used for this purpose, they are not efficient when a large number of such graphs are required for one particular performance attribute. In a schema/frame-based implementation, it is easier to represent them as procedural attachments known as demons. The following section presents an example of "DESIGN-CONTEXT" schema and describes how demons are used in establishing the performance requirements.

4.2.4 Example

The design context consists of user requirements and performance requirements as constraints to define the boundary of admissible design solutions. Figure 4.3 shows an example of "DESIGN CONTEXT" schema, its slots representing the design parameters, the source of information and the relationship between them. User input of building location is used to retrieve weather data corresponding to degree days and design temperatures and latitude. The building type and degree days information are used to establish the U_o for wall and roof in the slots "UO-WALL-HEATING" and "UO-ROOF" respectively. Once these U_o values are known, an energy budget (specifying the allowable heat loss through the envelope) can be established by multiplying the gross envelope area of components and the corresponding U_o requirements. If the building type is 4, then "OTTV-WALL" and "SOLAR-FACTOR" values are obtained from the ASHRAE 90A-1980 graphs to check the energy requirements for cooling season. These values are dependent on design latitude and are currently represented in demon functions written in commonLISP.

The relationship between performance attributes are dynamic in nature for maintaining the consistency of the design context information and the logical dependency of performance requirements. For example, when the value of energy budget is requested during the generation process, a demon is executed to operate on the U_o values

```
(defun u0-walls-a1-heating (degree-days)
     (cond   ((> degree-days 7800) 0.58)
             ((> degree-days 2500)
              (interpolate 2500 1.15 7800 0.58 degree-days))
             ((> degree-days 200)
              (interpolate 200 1.70 2500 1.15 degree-days))
             (t (print "*No ASHRAE requirements specified*"))
     )
)
```

Figure 4.2: Example of a LISP Function Representing ASHRAE Requirements

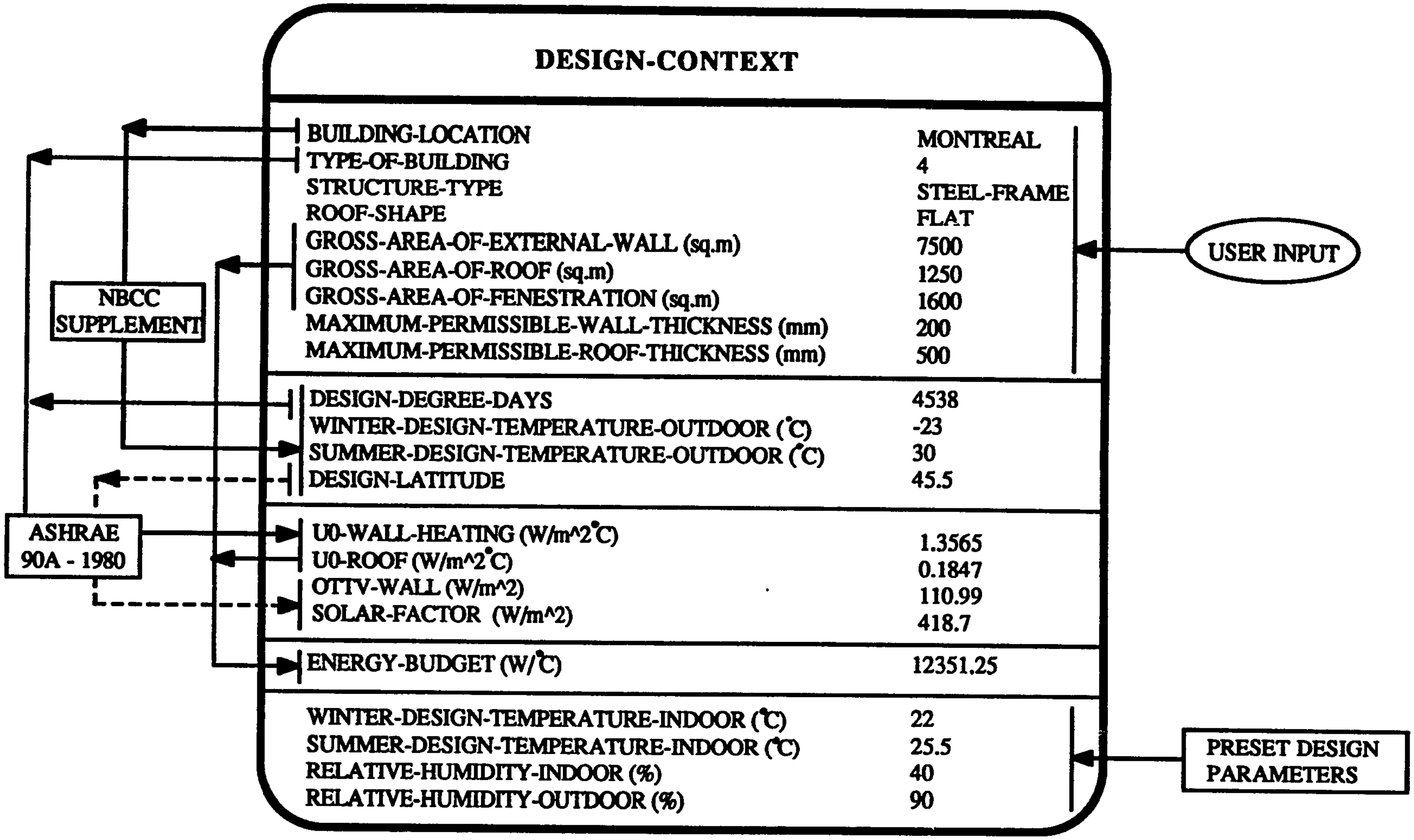


Figure 4.3: Example of a Design Context and Information Sources

and area of envelope components. The request for U_0 values in turn invokes a demon to calculate this value which depends on degree days. The data retrieval for degree days in turn executes a demon which uses the location name. Thus modifying any user input parameter such as building location or envelope area will automatically reassert all the performance requirements which are dependent on them.

The structure type and roof shape information are obtained as user input and used in the alternative generation to check the compatibility between systems and materials. In addition, the preset design parameters on indoor design temperatures and relative humidities are used in a condensation check for finding potential problems in wall systems as they are generated. Thus most of the performance related constraints established by the design context are used in the alternative generation process to check the suitability of materials, systems and their combinations.

4.3 Generation of design alternatives

Building envelope design alternatives typically consist of materials and construction types for the various layers of each component. The present study considers walls, roof and glazing systems in a hierarchy as described in Figure 2.5. The generation of design alternatives is a search process to identify the suitable combination of materials and construction types to meet the user and performance requirements of the design context. In order to achieve this, a knowledge base of information on material properties, construction types, compatibility between materials and systems, and design heuristics must be developed. An appropriate knowledge representation methodology for eflicent use of this information needs to be identified.

The generation of building envelope design alternatives can be approached from two levels of abstraction corresponding to materials and components. It is possible to represent material properties and their functional characteristics in a manner that each component be designed by identifying a suitable combination of materials. For example, in the case of roof system design, materials can be classified according to their functions such as deck, sheathing, insulation, waterproofing membrane, etc. Rules of design practice corresponding to different roof systems can be developed to find appropriate combination of materials meeting the performance requirements of the design context. This approach is found feasible at the component level for generating new design solutions [66]. But the applicability of this technique for the building envelope system is seriously limited by the fact that each component consists of a large number of layers and the number of possible combinations of materials and systems will increase exponentially. On the other hand if only component descriptions are used for generating the envelope assemblies, there is very limited data available and the number of alternatives generated will be too few to be of any use.

In order to develop a more practical and viable method for alternative generation, the present work considers the use of existing knowledge on standard construction types and at the same time treats structure and insulation as layers whose characteristics are not known initially but identified in the generation process. A number of constraints are checked during the generation of alternatives and this is achieved by representing building material properties and construction types information in a semantically related schema network. The details of knowledge representations and alternative generation process are described in the following paragraphs.

4.3.1 Building material properties

Building materials are grouped into three categories which have common attributes. All envelope materials excepting insulation and glazing are represented consisting of properties such as thermal resistance, vapour resistance, density and cost. Figure 4.4 shows the details of a schema "BUILDING-MATERIALS" and an example of instances of this schema representing the material properties of concrete, brick and wood sheathing. There are about forty such building materials currently available in the BEADS system. Figure 4.5 shows the instance names of generic building materials present in the knowledge base. The material properties for these instances are obtained from several different sources ([14], [65], [67], [68]). Each of these building materials can be used to define the various layers in a wall or roof system. Building materials which may be used as structural members are linked to a schema "STRUCTURAL-MEMBER". This schema provides two additional slots for specifying the knowledge about the suitability of the structural member in a particular design context with respect to building type and structure type. Example of structural materials definitions are shown in Figure 4.6. The suitable structure type and building type information for various structural materials are obtained from a handbook on preliminary design [69].

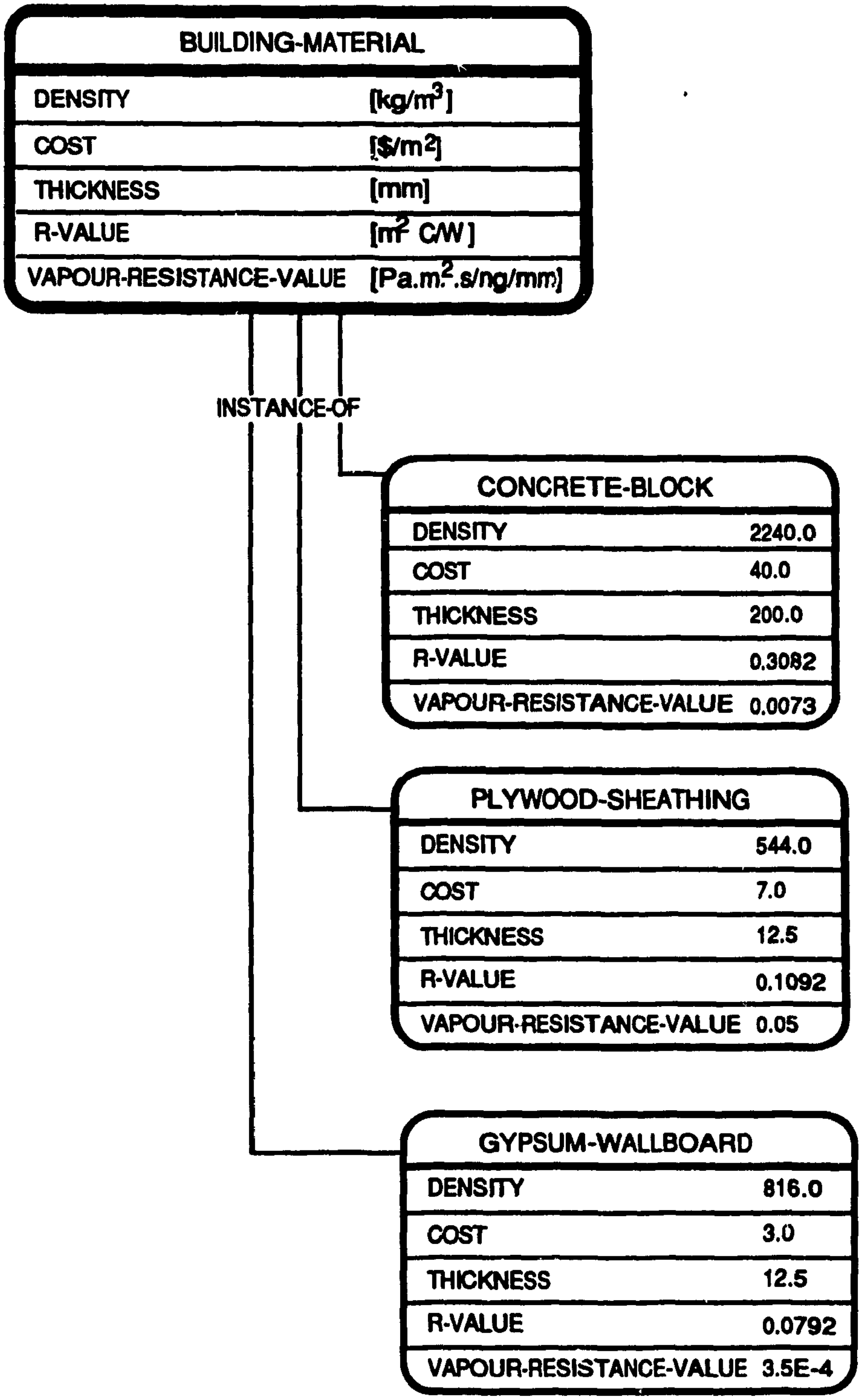


Figure 4.4: Building Materials Description

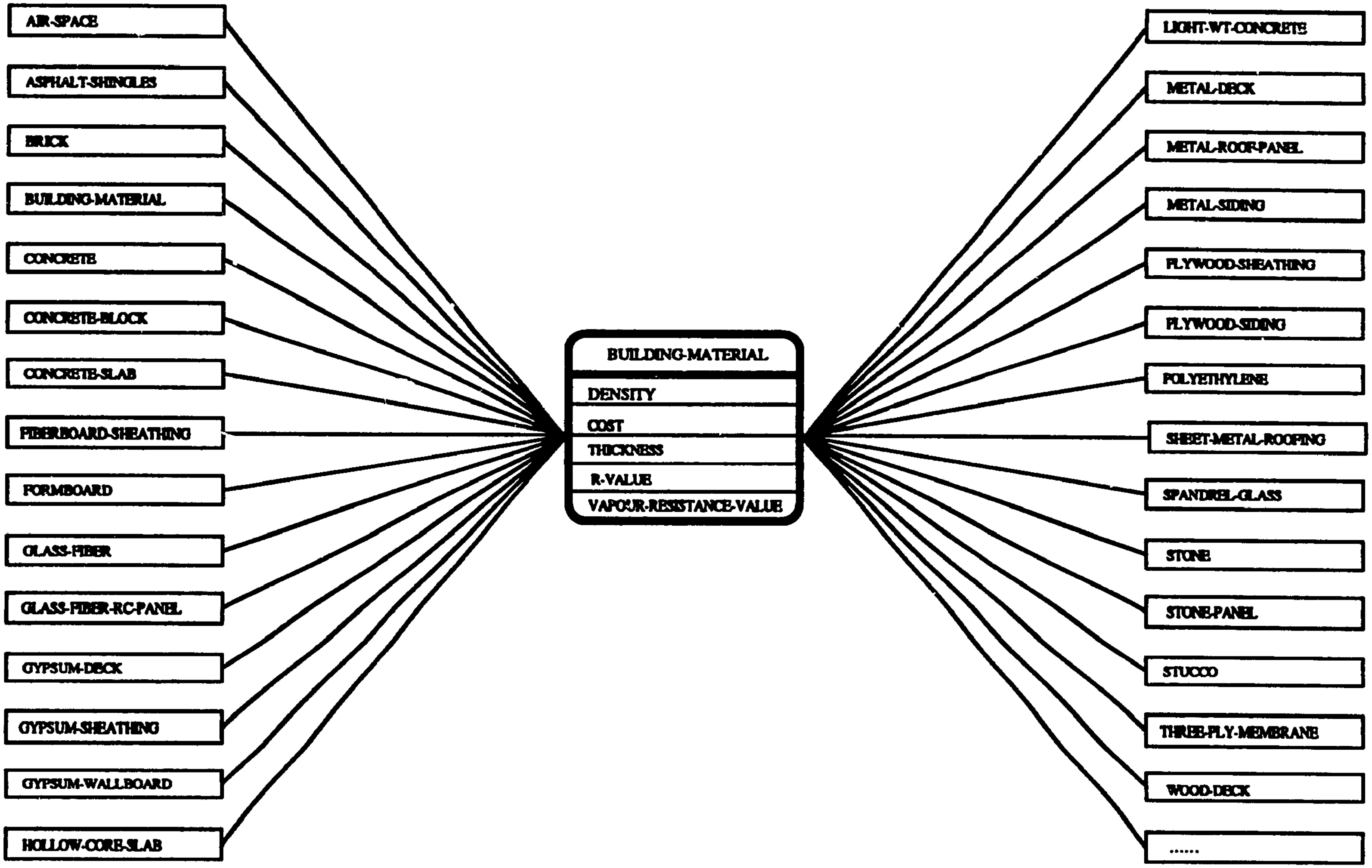


Figure 4.5: Instances of "BUILDING-MATERIAL" Schema

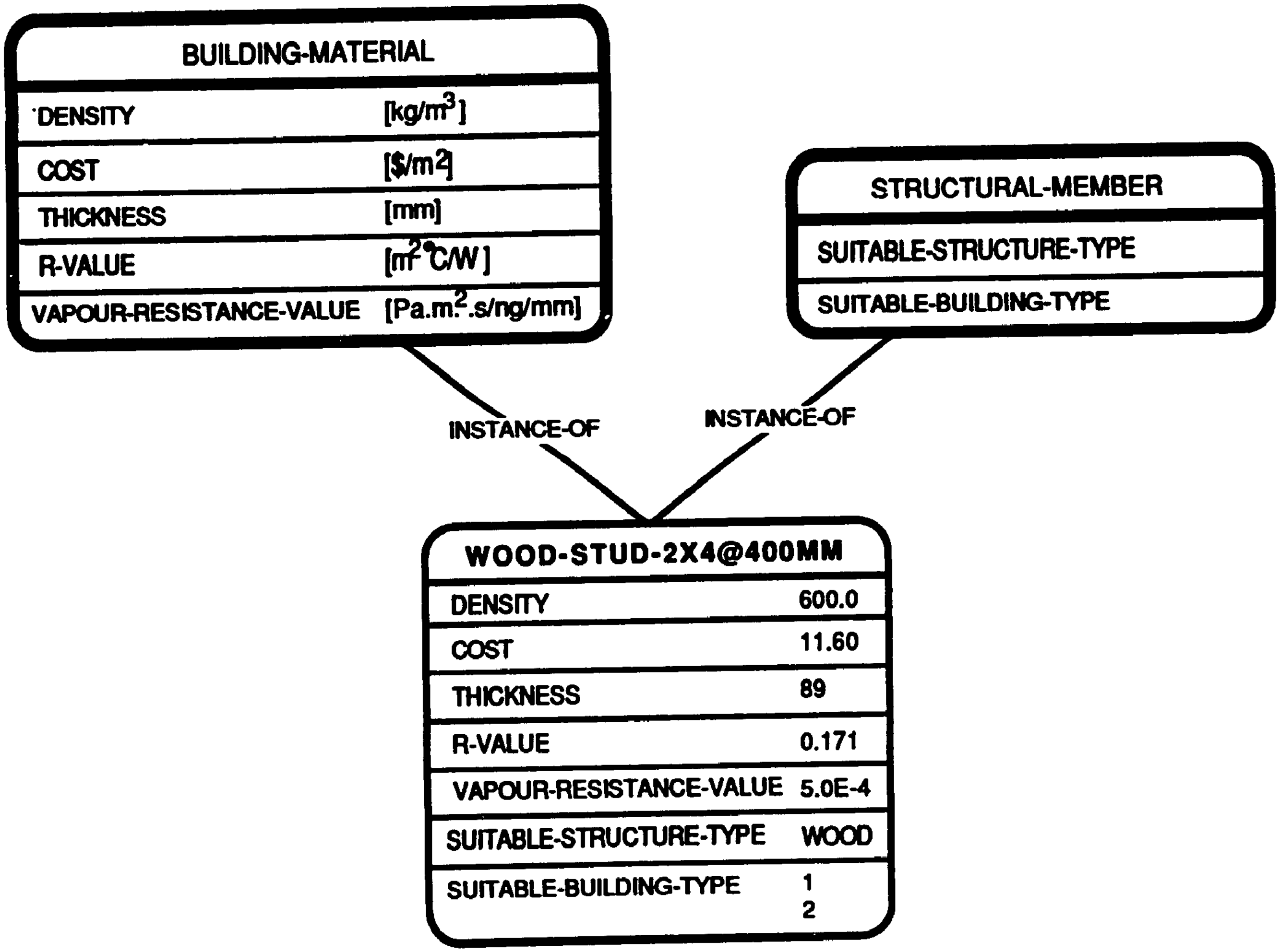


Figure 4.6: Structural Material Description

Insulation materials are available in many different standard thicknesses and the use of a particular insulation type needs a careful evaluation as to the performance of the wall or roof for which it is used. Though insulation materials are very similar to the building materials discussed above, there are specific differences in terms of defining compatibility and suitability to envelope components and materials. Figure 4.7 shows an example of "INSULATION-MATERIAL" schema and its instance representing the properties of expanded polystyrene. The "INSULATION-MATERIAL" schema consists of slots corresponding to available thicknesses, thermal resistance, vapour resistance, density, cost and finally a slot for the building material names which are incompatible for use with this particular insulation. The list of values present in the slots "AVAILABLE-THICKNESS" and "COST" have one to one correspondence. Once a particular thickness of insulation is found suitable in a design context, then the appropriate insulation cost is retrieved for calculating the total cost of component. Each instance of insulation material is linked to a parent schema corresponding either to "WALL-INSULATION" or "ROOF-INSULATION" thereby semantically specifying the envelope component in which this insulation material can be used. There is a total of about six insulation types as shown in Figure 4.8 that are currently available in the BEADS system. Design practice and guidelines as to the incompatible materials for an insulation material can be obtained from technical literature [70]. The available thicknesses and cost are obtained from the Yardsticks for Costing handbook [68] and the properties can be found from the references cited earlier for the building material properties data.

Glazings have properties which are different from other building materials. For example, the shading coefficient and compatibility information are specific to glazings. In the BEADS system, glazings are represented describing the information required both as a material and as an envelope component. The "GLAZING-TYPE" schema consists of slots corresponding to cost, shading coefficient, R-value and suitable building type information. The shading coefficient of glazings is used during cooling check for the envelope. The details of "GLAZING-TYPE" schema and its instances are shown in Figure 4.9. The properties for all these glazing types are obtained from a technical guide on window design [71]. Some of the information and knowledge for specifying the suitable building type are derived from interviews with an architect [72].

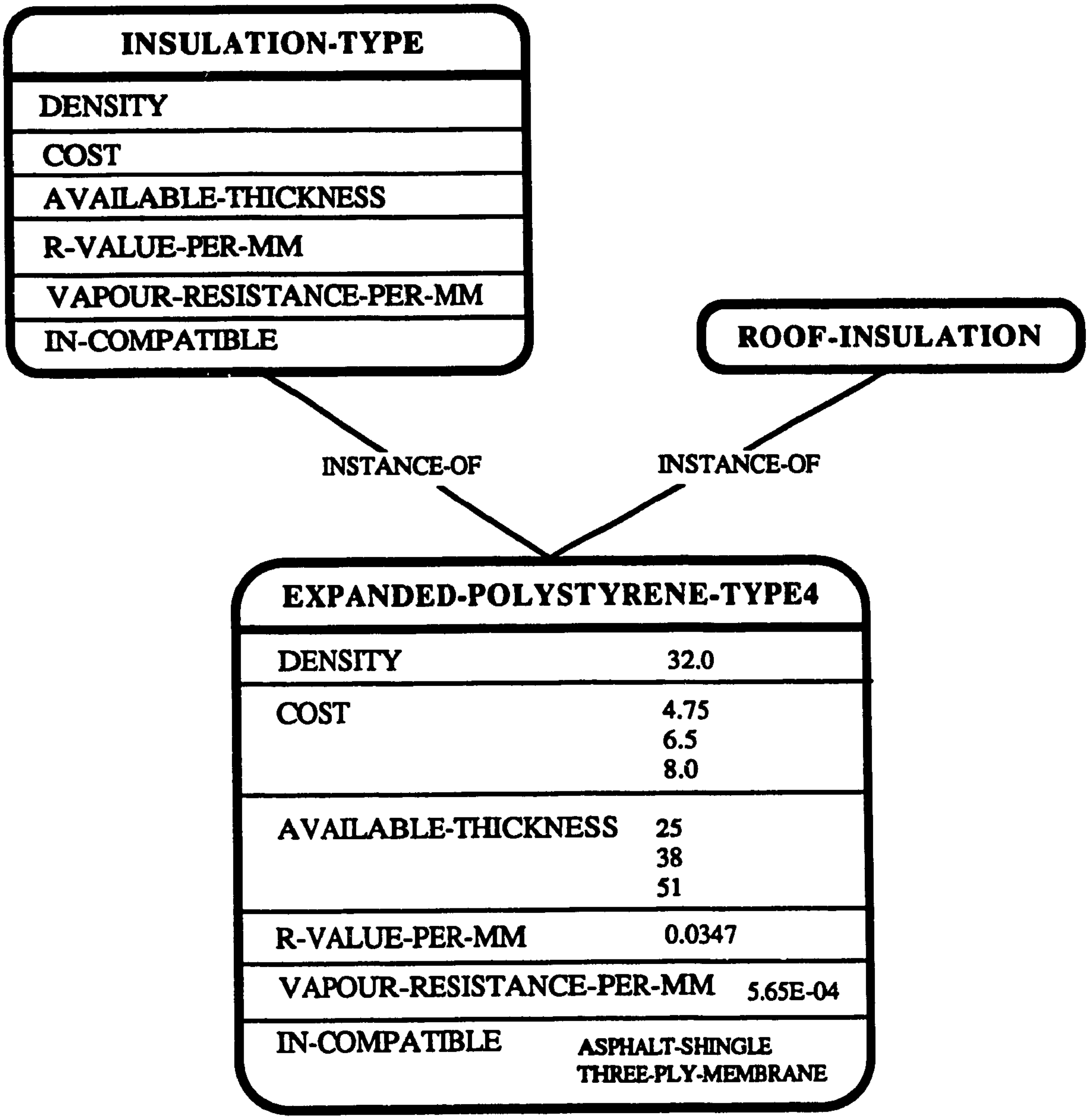


Figure 4.7: Insulation Material Description

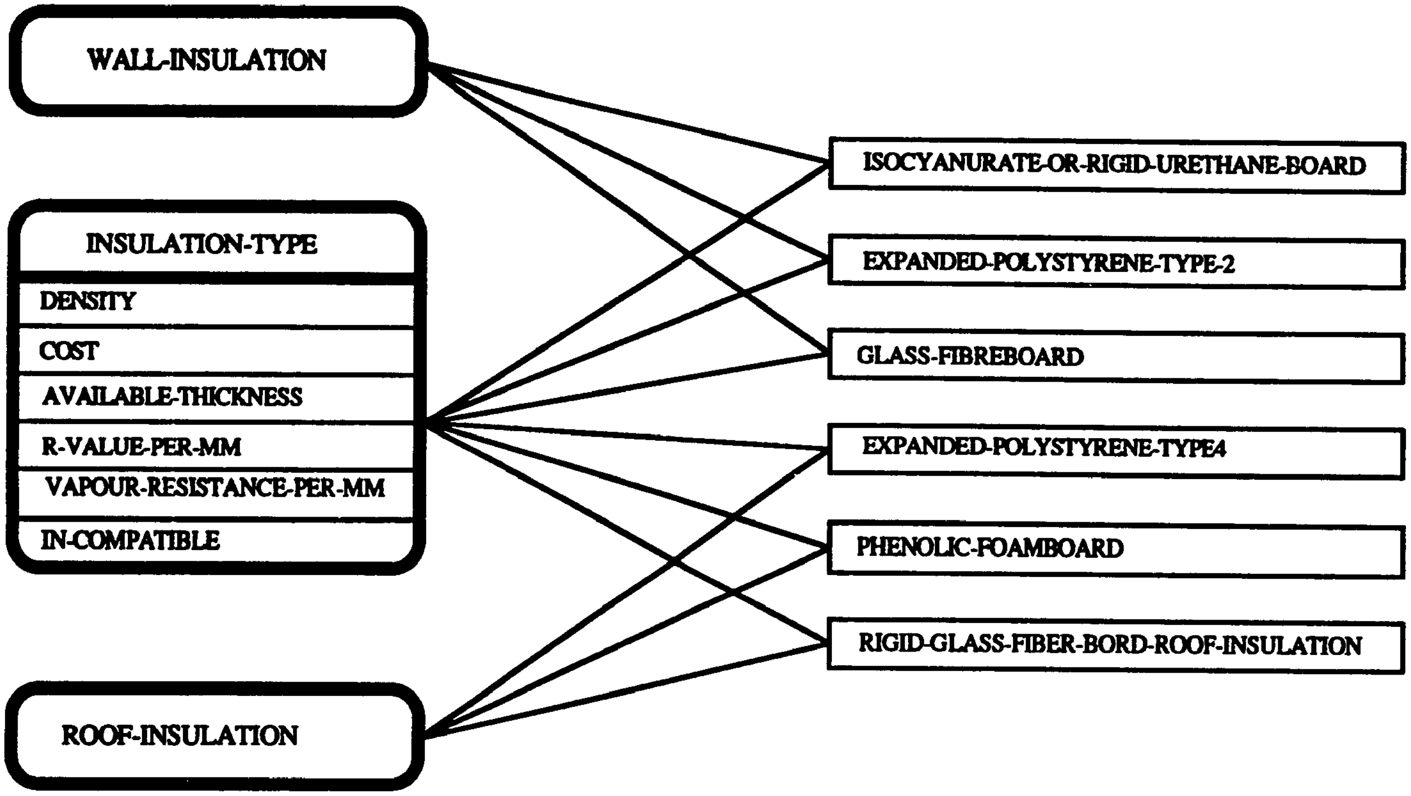


Figure 4.8: Instances of Wall and Roof Insulation Materials

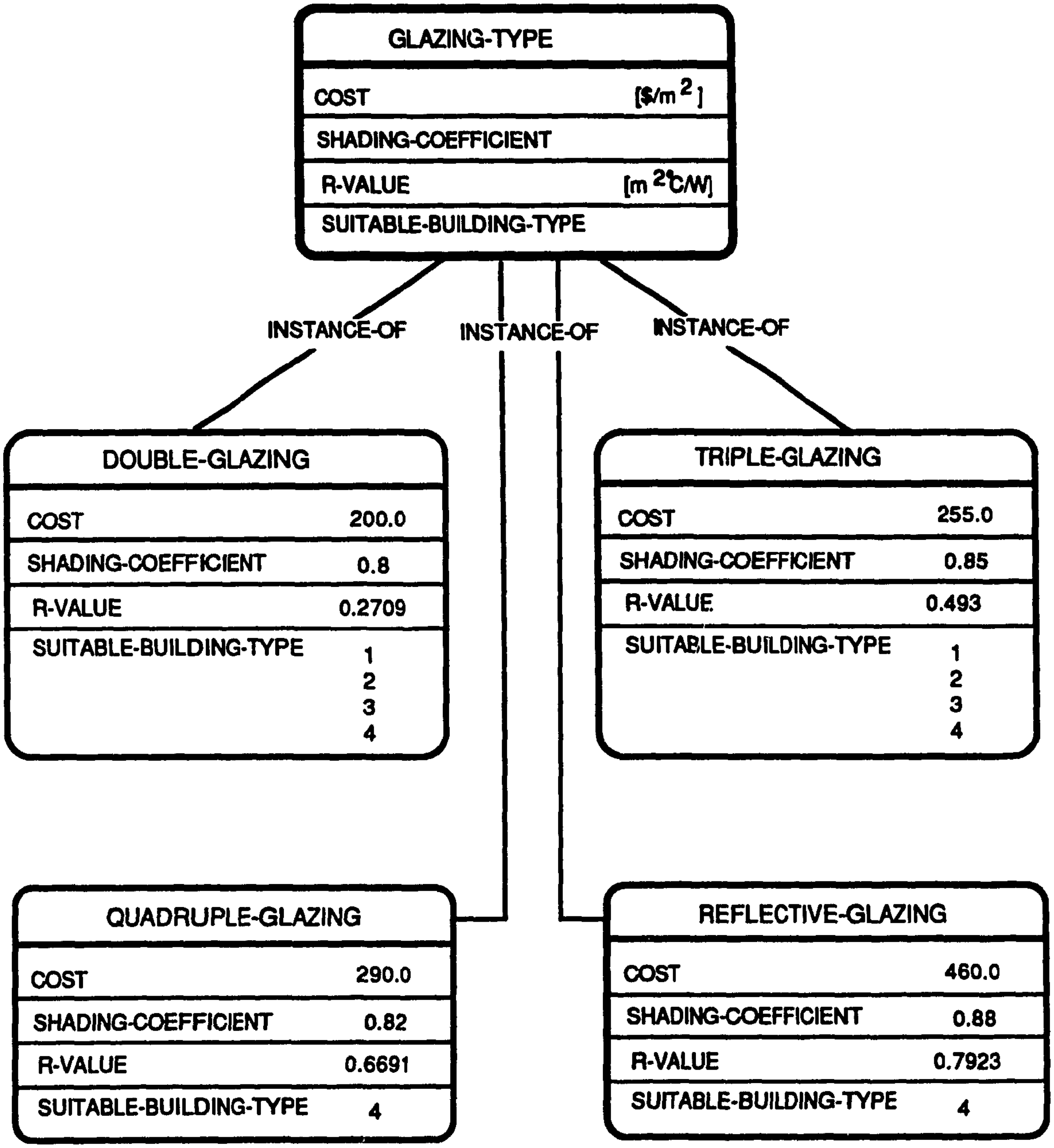
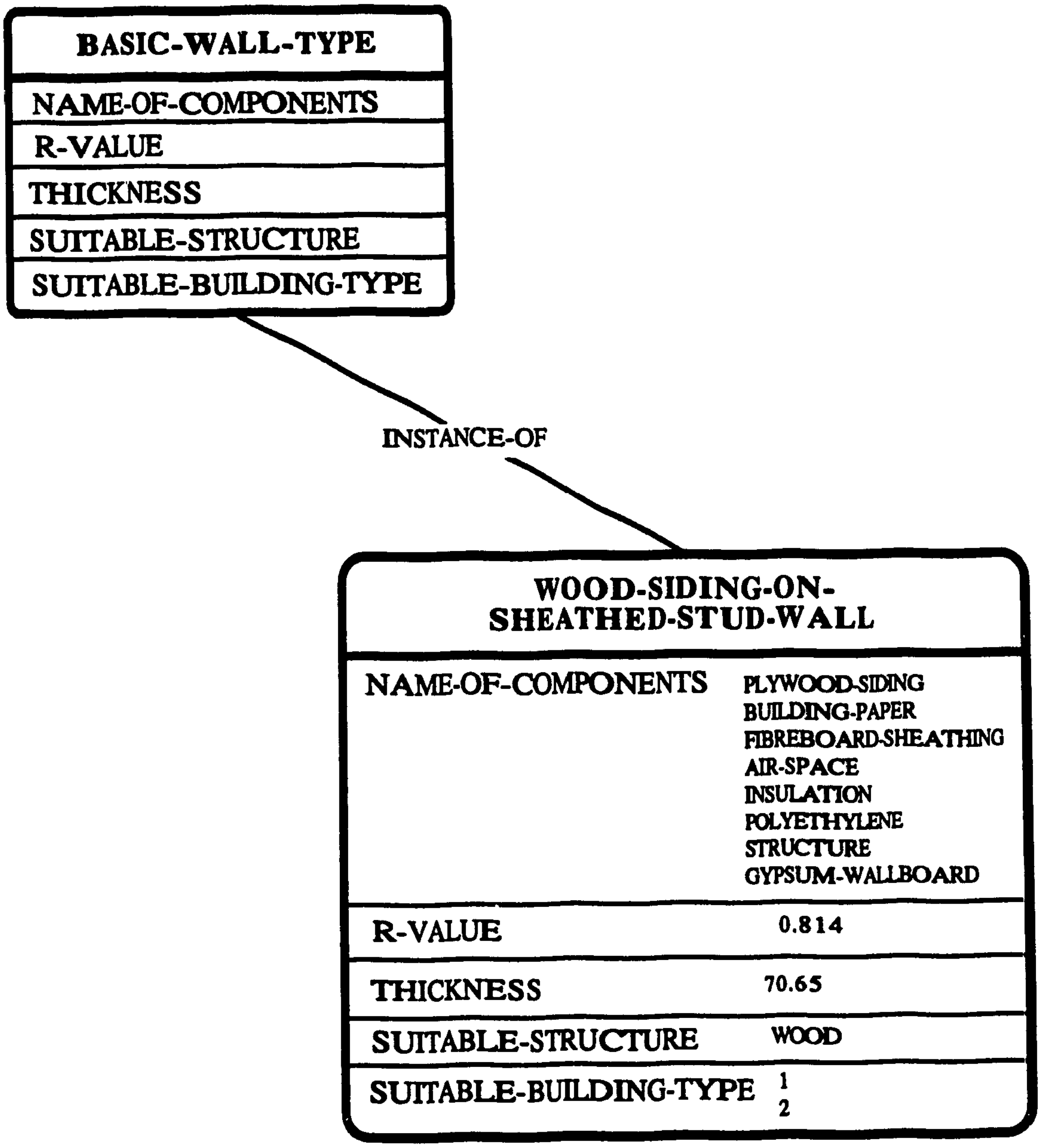


Figure 4.9: Glazing Type Description

4.3.2 Construction Types Information

There are many standard basic wall and roof types available to the designer from design handbooks and product literatures. For example, the American Institute of Architects, in a handbook on energy efficient design [65], has provided a list of basic wall and roof types that are commonly used in practice. These basic construction types provide the description of materials for the various layers other than insulation. During design, one can select a particular basic wall type and find a suitable insulation material to meet the thermal resistance and other requirements of the wall assembly. This approach enables the designer to use well researched construction types which can also satisfy the requirements of different design contexts. But the use of such basic construction types are constrained by the type of structural system and building occupancy, etc. In order to represent these descriptions, two schemas "BASIC-WALL-TYPE", and "BASIC-ROOF-TYPE" are defined.

Figure 4.10 shows the "BASIC-WALL-TYPE" schema and an example of its instance with details of "WOOD-SIDING-ON-SHEATHED-STUD-WALL". The "NAME-OF-COMPONENTS" slot describes the various layers present in this basic wall type. The order of values in this slot correspond to that of the layers from exterior to the interior of the wall. All values in the "NAME-OF-COMPONENTS" slot correspond to schema names of building materials, excepting for "INSULATION" and "STRUCTURE". For example, wood siding on sheathed stud wall consists of plywood siding, building paper, fibreboard sheathing, air cavity, polyethylene vapour barrier and gypsum wall board which are instances of "BUILDING-MATERIALS" schema. During the generation of design alternatives, the value "INSULATION" is replaced with a suitable insulation material and the value "STRUCTURE" is replaced with a suitable structural material. The values for "THICKNESS" and "R-VALUE" slots are redefined by adding corresponding values retrieved from the appropriate instances of "BUILDING-MATERIAL" schema. The thickness and thermal resistance of the basic wall are used in determining the suitability of this basic wall to meet the constraints of permissible wall thickness and U_0 for wall heating established by the design context. Further, the "SUITABLE-STRUCTURE" and "SUITABLE-BUILDING-TYPE" slots represent the information as to the suitability of this wall type in a given context relating to the user input on structure type and occupancy. These are essentially constraints used in the alternative generation process to eliminate or considering a particular basic wall type. There are seventeen basic wall types present in the BEADS system. Figure 4.11 shows the instance names and Appendix A lists the



BASIC-WALL-TYPE
NAME-OF-COMPONENTS
R-VALUE
THICKNESS
SUITABLE-STRUCTURE
SUITABLE-BUILDING-TYPE

WOOD-SIDING-ON-SHEATHED-STUD-WALL	
NAME-OF-COMPONENTS	PLYWOOD-SIDING BUILDING-PAPER FIBREBOARD-SHEATHING AIR-SPACE INSULATION POLYETHYLENE STRUCTURE GYPSUM-WALLBOARD
R-VALUE	0.814
THICKNESS	70.65
SUITABLE-STRUCTURE	WOOD
SUITABLE-BUILDING-TYPE	1 2

Figure 4.10: Basic Wall Type Description

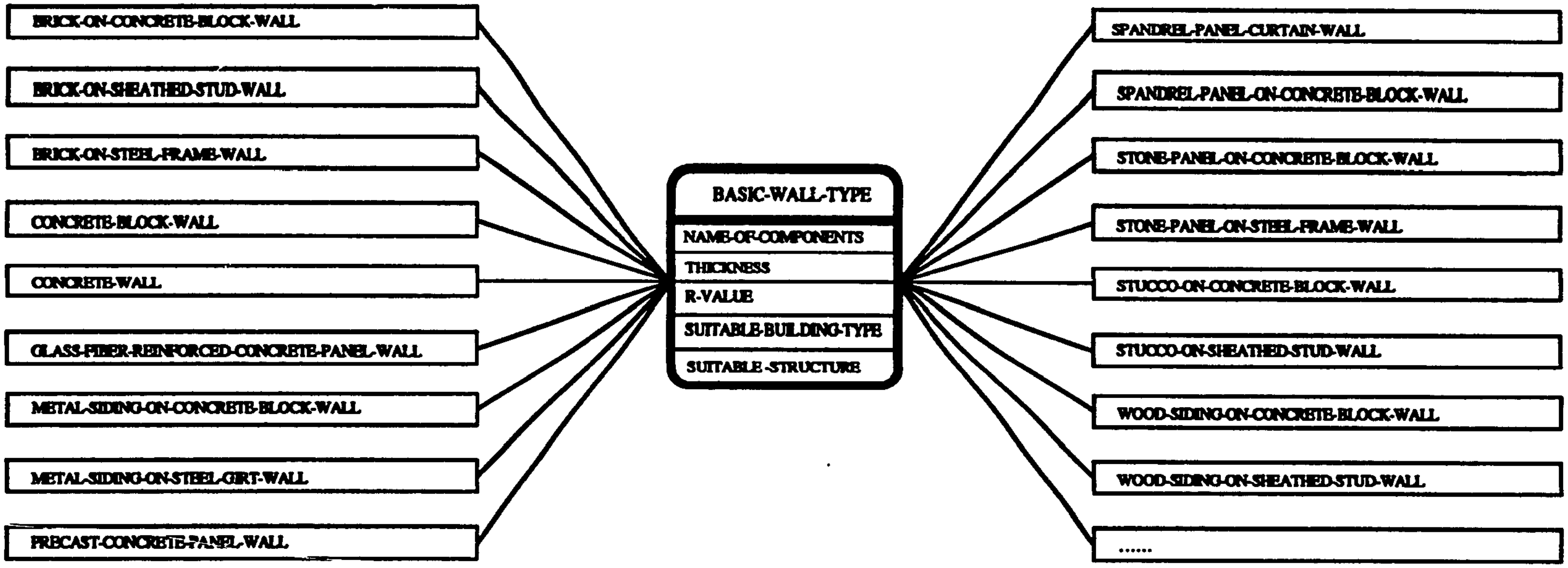


Figure 4.11: Instances of "BASIC-WALL-TYPE" Schema

schema definitions for these basic wall types. Information on construction details are obtained from a handbook on energy efficient design [65]. Heuristic information on suitability of each type for building occupancy and structure type are provided by a practising architect [72].

Similar to basic wall types, there are many basic roof types developed and widely used. The description of such roof types and knowledge about their suitability can be represented in schemas. The "BASIC-ROOF-TYPE" schema shown in Figure 4.12 consists of attributes corresponding to the details of construction, suitable structure type, building occupancy, thermal resistance value and total thickness. All these slots have the same role as discussed above in representing the basic wall type information. There are ten basic roof types currently present in the BEADS system and Figure 4.13 shows the names of these basic roof types. Schema definitions for these roof types are listed in Appendix A.

The representation of material properties and construction types information in schema form has many advantages in terms of semantic relationships and encoding heuristic knowledge. For example, insulation types and their material properties can be described in a generic form, but by relating them to be instances of a schema of wall insultation, the use of this insulation material is semantically specified as suitable only for wall assemblies. There is also the advantage of describing construction types in terms of the constituent materials, representing materials in terms of their properties and establishing the properties of construction types automatically based on the semantic relationship.

Figure 4.14 shows how this semantic relationship between construction types and material properties information are resolved during the generation process. "CONCRETE-BLOCK-WALL" is a basic wall type consisting of the following building materials: "CONCRETE-BLOCK", "POLYETHYLENE" and "GYPSUM-WALLBOARD". The thickness and R-value of the basic wall are derived from the description of individual building materials. The value "INSULATION" in the "NAME-OF-COMPONENTS" slot of the wall description will be replaced with an appropriate insulation material such as "EXPANDED-POLYSTYRENE-TYPE2" which is an instance of "INSULATION-TYPE" and "WALL-INUSLATION". A suitable thickness of insulation satisfying the design constraints will be chosen and the insulation material properties will be added to the wall description, thus transforming a basic wall type to be a feasible wall type with

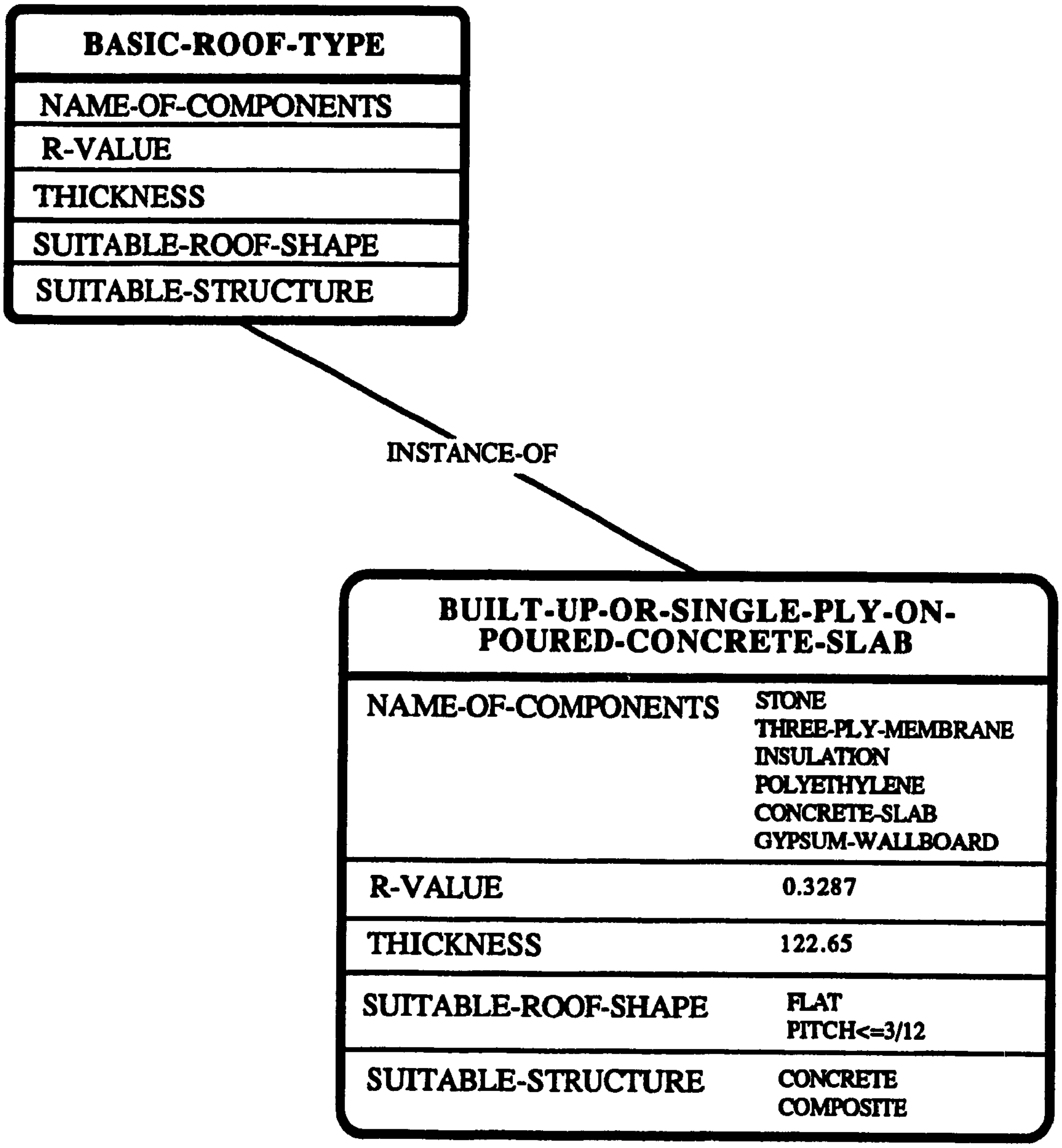


Figure 4.12: Basic Roof Type Description

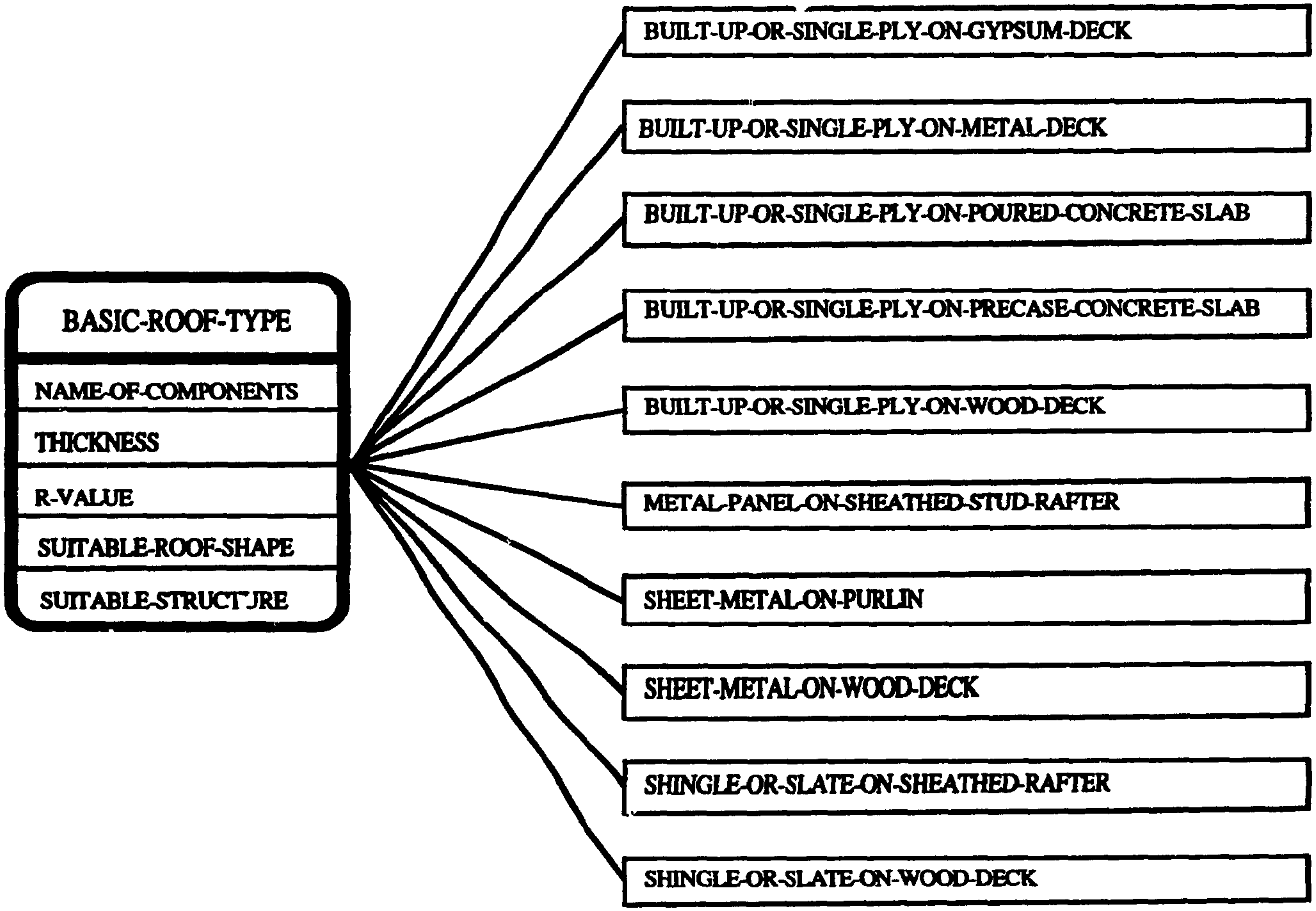


Figure 4.13: Instances of "BASIC-ROOF-TYPE" Schema

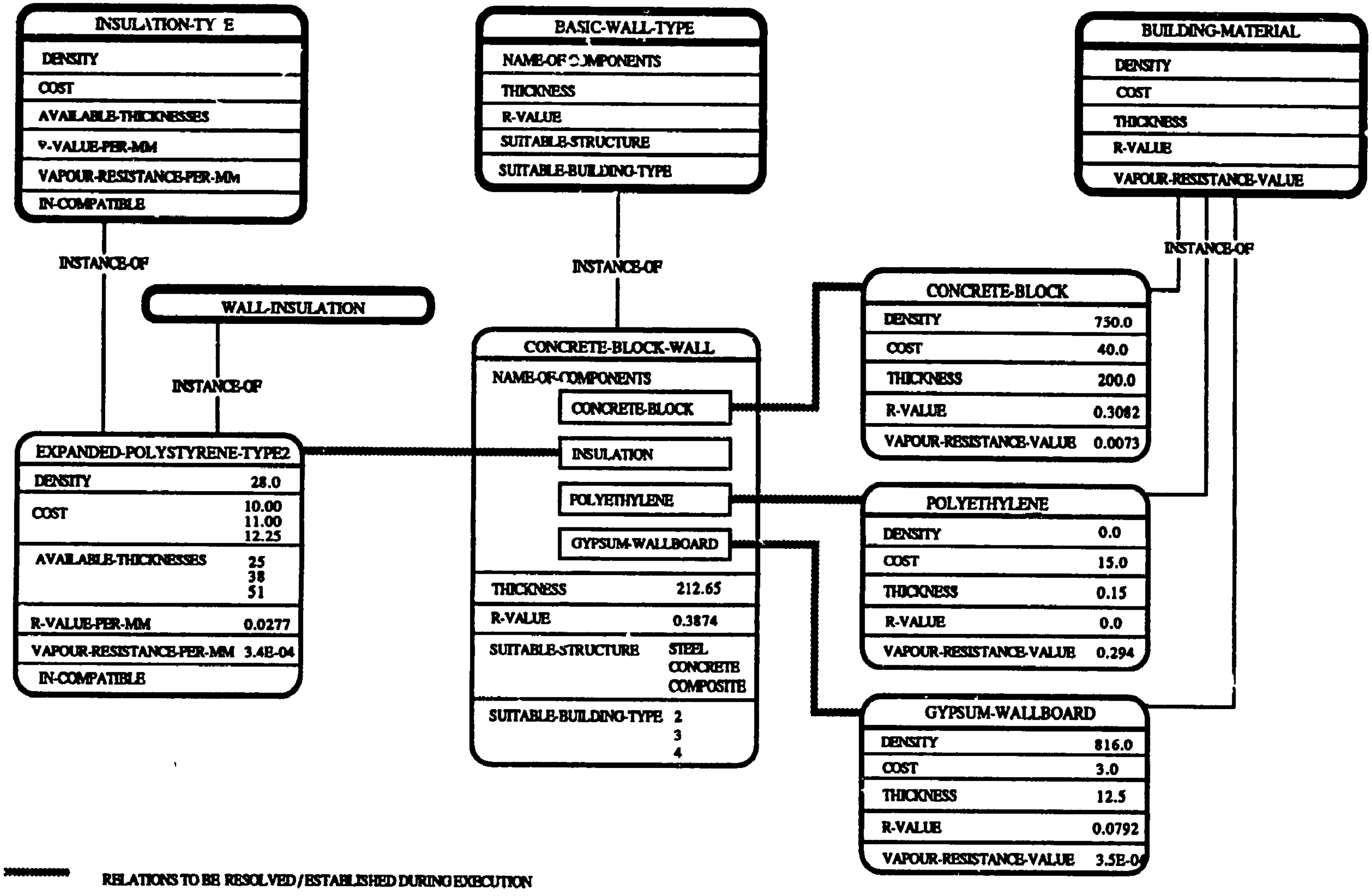


Figure 4.14: Example of Relationship Between the Various Schemas

complete details. As the design proceeds, other insulation materials and thicknesses will be considered to generate more design alternatives. A similar strategy is employed for generating descriptions of feasible roof systems.

Heuristic knowledge with respect to compatibility between materials and suitable structure types corresponding to construction types, etc are represented as slot values in the present study. This is very much different from the traditional rule-based representation in which they are represented as condition-action relations. The primary reason for describing them in the schemas is to keep this knowledge in a structured form for easy retrieval and maintenance. In addition, this representation is essential for implementing an efficient search strategy for generating feasible design alternatives.

4.3.3 Alternative Generation Process

The generation of design alternatives is viewed as a constraint-based search process in which constraint satisfaction is checked at three different levels, namely: material, component and system levels. Constraints for the alternative generation process are defined by user and performance requirements, functional relationships between materials and components, and design heuristics. At the material level, only material compatibility constraints are considered. But at the component and system levels many other constraints are checked. These checks may be simple predicate relationships or more complicated analytical computations and verifications. In the BEADS prototype implementation, the following constraints are considered:

(i) Compatibility of structure type for wall and roof
(ii) Compatibility of building type for wall, roof and fenestration
(iii) Permissible thickness of wall
(iv) Minimum R-value requirement for wall assembly
(v) Static condensation check for wall assembly
(vi) Compatibility between roof shape and roof type
(vii) Permissible thickness of roof
(viii) Energy consumption of the envelope
(ix) Cooling check for wall assembly, if required
(x) Compatibility of materials

Figure 4.15 presents a flow chart of the various tasks and their sequence during the generation process. The generation of design alternatives begins with the selection of a basic wall type which is suitable for the user specified structure type, building type and maximum permissible thickness. If this basic wall type requires a structural framing, then an appropriate structural material is identified and the basic wall description is redefined. Once such a basic wall type is identified, the available wall insulation materials and their thicknesses are considered one after the other to meet the thickness requirement and total thermal resistance of the wall assembly including the insulation. Now this wall assembly is checked for possibility of interstitial condensation. A simple static condensation analysis (considering only moisture diffusion) is carried out by knowing the thermal and vapour resistance of each layer, the indoor and outdoor design temperatures and relative humidities. Thermal and vapour pressure profiles are established to determine the location of dew point. If the dew point lies in the wall cross section, then this alternative is eliminated. A detailed discussion of this condensation check method is reported elsewhere by Fazio and Gowri [73]. If the condensation check is passed, then the generation process proceeds to identify a suitable basic roof type and then a roof insulation. The above process of generating a design alternative description may be referred to as prototype identification, retrieval and refinement [33].

In selecting the basic roof type once again the constraints on suitability of structure type and building type, and the permissible roof thickness are applied. In selecting roof insulations, material compatibility is also ensured. Now the wall and roof assembly are combined with a suitable glazing type to form an envelope assembly. Energy consumption reflecting the heat loss through the present combination of envelope components is calculated by multiplying thermal transmittance and surface area for wall, roof and fenestration. This value is checked against the energy budget established by the design context. If the building requires a cooling check (according to ASHRAE standard 90A-80), then this is also carried out. If all these checks are successful in meeting the requirements, then this combination of envelope components is specified in an instance of feasible alternative description as shown in figure 4.16. Such instances are created at run time and are related to the schema "FEASIBLE-ALTERNATIVE" which consists of slots for representing basic wall type, wall insulation, basic roof type, roof insulation, glazing type, thicknesses of walls, roof and insulations, and the performance attributes and utility values required for ranking the alternatives. The energy consumption, total thickness of wall and roof are values of constraints evaluated during the generation process. But the material cost of the envelope assembly is calculated at

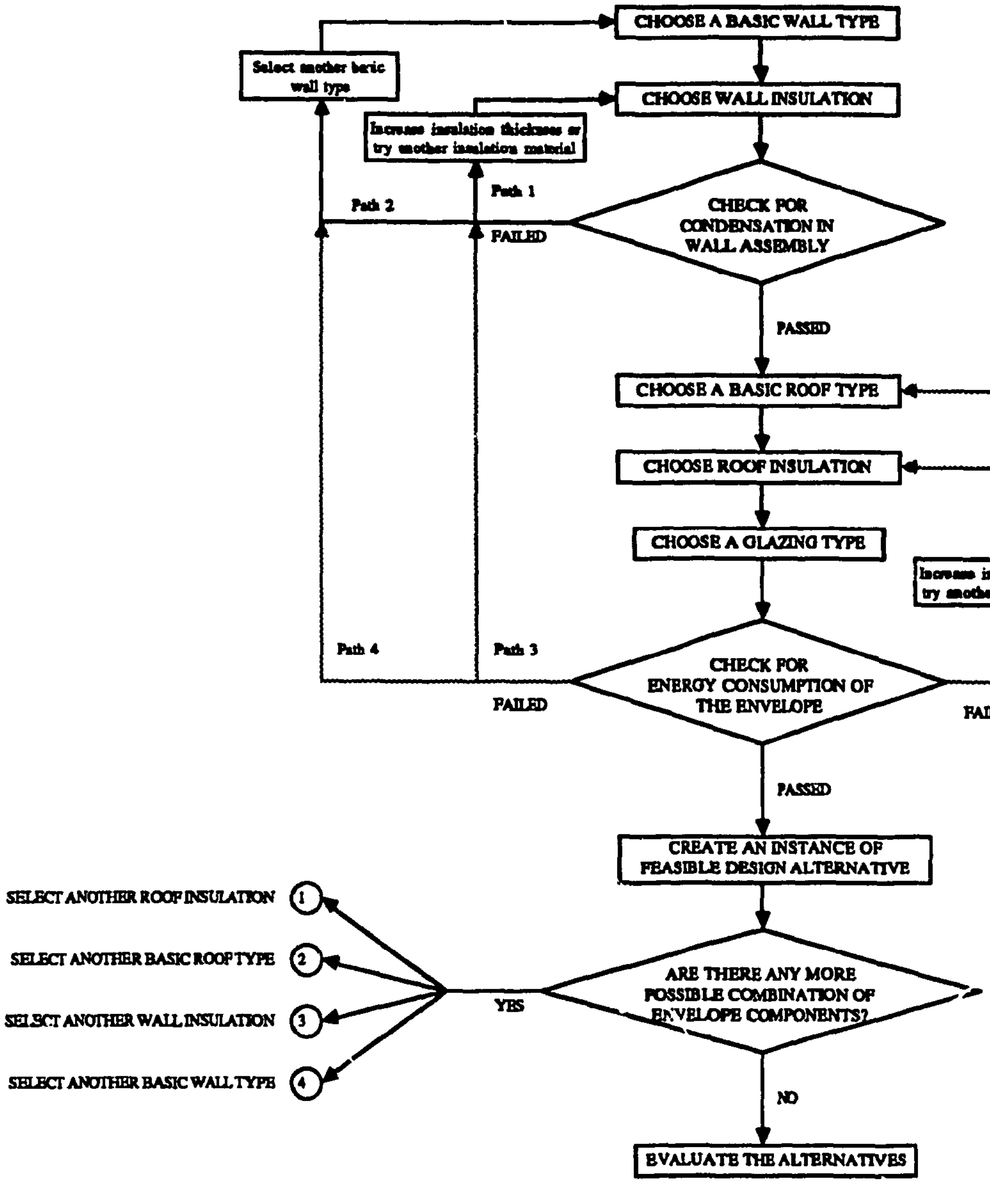


Figure 4.15: Flow chart of tasks in generating design alternatives

FEASIBLE-ALTERNATIVE-54		
WALL-TYPE	STUCCO-ON-CONCRETE-WALL	
ROOF-TYPE	BUILT-UP-OR-SINGLE-PLY-ON-METAL-DECK	
INSULATION-TYPE-FOR-WALL	GLASS-FIBRE-BOARD	
INSULATION-TYPE-FOR-ROOF	RIGID-GLASS-FIBRE	
GLAZING-TYPE	DOUBLE-GLAZING	
ENERGY-CONSUMPTION	2701.26	(W/ °C)
MATERIAL-COST	594982.0	($)
TOTAL-THICKNESS-OF-WALL	281.5	(mm)
TOTAL-THICKNESS-OF-ROOF	140.0	(mm)
UTILITY-VALUE	32.3 9.8 1.4 10.6 10.5	

Figure 4.16: Instance of a "FEASIBLE-ALTERNATIVE" Schema

the time of defining a feasible alternative. This material cost is provided by the building materials description and are in-place costs used for preliminary estimates. These performance attributes will be used in the ranking and selection of alternatives.

The above process of alternative generation is repeated until all feasible combination of envelope components are identified. At each instant of constraint violation, the alternative generator back tracks to the previous level. For example, if the condensation check for a wall assembly fails, then other available thicknesses of the insulation material are tried until a successful one is found. If there are no suitable thicknesses in this insulation material, then another wall insulation material is chosen and the process is continued. On the other hand, if there are no other wall insulations available, then a different basic wall type is considered and the subsequent checks are made. This process follows a top-down search process for generating all feasible combinations of components and materials by satisfying constraints at various levels.

4.4 Ranking and Selection of Alternatives

The alternative generation process results in the identification of all feasible combinations of envelope components which meet the user and performance requirements specified by the design context. These feasible alternatives must be ranked in a systematic fashion for enabling the designer to select the most suitable solution. A satisfying solution for one designer may be clearly unsatisfactory for another. Hence it is evident that the overall strategy employed by the selection mechanism should allow a high degree of flexibility in the ways in which it may be used. In addition, each alternative has different levels of performance for each attribute which must be considered in the selection process. One fundamental difficulty here is that each performance attribute has a different unit of measure and this needs to be normalized for comparing the overall performance of alternatives because acceptability will not depend on an individual attribute but on overall quality [41].

The BEADS system addresses this issue of flexibility for selection in two stages. The first one allows the designer to specify the preferred construction types for wall and roof, and the second stage lets the designer specify priorities of performance attributes. The preferred feasible alternatives are ranked on a percentile scale for each performance attribute and the priorities on performance attributes are used to compute the overall utility value for the alternatives. This approach provides a simple, yet meaningful scheme for decision-making at the preliminary design stage, since it relies on the best relative performance of a given alternative within a set of feasible ones. Though there are many optimization techniques available for the selection process, the present study purposely avoids the use of these techniques based on the premise that they are not practical for use at the earliest stages of design when only few parameters are known to an acceptable degree of accuracy. Besides these techniques require extensive input data, understanding of the mathematical basis for optimization and do not allow the designer intervention during the selection process.

During the alternative generation process, two lists of names corresponding to basic wall and roof types are created. These lists are presented to the user for specifying the preferred construction types. The design decision space is then narrowed down considerably and consists only of alternatives representing the users choice of construction types. Ranking of these alternatives are done by considering the relative performance for each one of the following attributes:

(i) Energy consumption
(ii) Material cost
(iii) Total wall thickness
(iv) Total roof thickness

The performance corresponding to each of the above attributes and for all the feasible alternatives are determined at the time of generation. The performance data for each attribute has a different unit and needs to be normalized for ranking purposes. In order to address this issue, utility values based on a percentile scale can be used. It is possible to identify the alternatives with the most and least preferred performances and they can be specified to have maximum and minimum utility values respectively. All other alternatives can now be assigned utility values based on a linear interpolation scheme reflecting the relative performance. For example, the design alternative corresponding to minimum energy consumption is given a utility value of 100 and that with maximum energy consumption is assigned a utility value of unity. All other feasible alternatives are assigned utility values for energy consumption using a simple interpolation scheme, and these values will lie between 1 and 100. In a similar fashion, utility values for other performance attributes are computed with the objective being minimum material cost, minimum overall thicknesses for wall and roof.

An additive utility criteria is then used to consolidate the utility values of all performance attributes, thus establishing an overall utility value for each alternative. At this time, it is possible to accommodate the designer priorities on performance attributes. These priorities can be treated as weights with which the utility values can be scaled before calculating the overall utility value. Finally, the alternatives are ranked according to overall utility values and the alternative with the highest overall utility value is suggested as the most suitable solution.

CHAPTER 5

IMPLEMENTATION AND VALIDATION

5.1 Introduction

The proposed design methodology and knowledge representation have been implemented in a prototype system known as BEADS. This prototype implementation aims at the selection of external walls, roof and glazings of a building envelope by taking into account the performance attributes relating to energy efficiency, structural and material compatibilities. The software architecture and programming aspects of BEADS implementation are presented in this chapter. In order to examine the validity of the prototype knowledge base, two practical design problems have been solved using BEADS. The results are compared to the original design proposals.

5.2 Software Architecture

The prototype system comprises of three components namely knowledge base, inference mechanism and user-interface which are typical of any knowledge-based system. Figure 5.1 presents a graphical layout of the software architecture and modules that are currently implemented. The information flow and control flow between the various modules are also identified in this figure. KNOWLEDGECRAFT in a VAX-VMS environment is used for implementing the BEADS prototype.

5.2.1 Knowledge Base

The knowledge base consists of information on ASHRAE Standard performance requirements, weather data, material properties, construction type descriptions, heuristic knowledge and semantic relationships between envelope components and materials. The knowledge base provides all the information required in the various stages of the design process. The context manager uses the weather data and ASHRAE

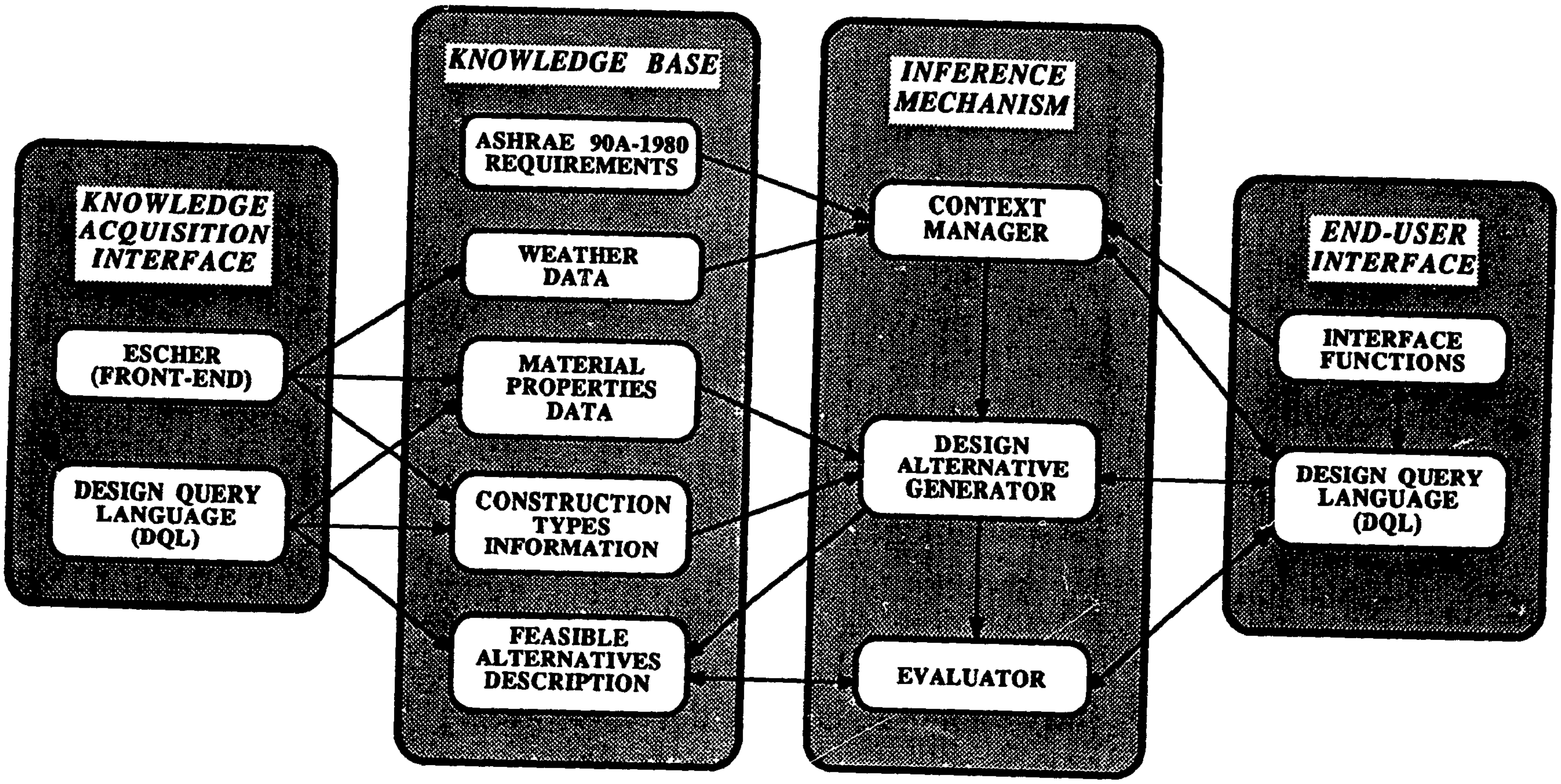


Figure 5.1: Software Architecture of BEADS Prototype Implementation

requirements to define the quantitative performance requirements for establishing the design context. The material properties and construction type descriptions are used by the alternative generator during the search for feasible design alternatives. The description of feasible design alternatives, once identified are temporarily added to the knowledge base and this information is used by the evaluator for ranking and final selection.

The various types of information present in the knowledge base are implemented in the form of commonLISP functions and CRL schemas. The details of knowledge representation have been described in the previous chapter. ASHRAE requirements are represented as commonLISP functions (Figure 4.2) and are created using a standard text editor. These functions can be easily edited for updating and maintenance. The CRL schemas representing the weather data, material properties and construction type information are defined using ESCHER, a front-end to KNOWLEDGE-CRAFT [53]. The knowledge base maintenance for these schemas can be carried out using either ESCHER or a design query language (DQL) developed specifically for use with BEADS. Details of DQL features and implementation are described later in this chapter.

5.2.2 Inference Mechanism

The BEADS inferencing process is carried out by three modules corresponding to the establishment of the design context, generation of alternatives, ranking and selection. The context manager obtains input from the user and retrieves all the required information from the knowledge base before defining the quantitative requirements. The "DESIGN-CONTEXT" schema described in the previous chapter (Figure 4.3) shows the various attributes and the relations between them. These relationships are implemented as demons for ensuring the dynamic and logically dependent nature of the attributes. For example, the value in "U0-WALL-HEATING" slot of the "DESIGN-CONTEXT" schema is obtained by knowing the value of "DESIGN-DEGREE-DAYS" slot which in turn is dependent on the value of the "BUILDING-LOCATION" slot. Demons are implemented by defining the demon name and function name in appropriate schemas corresponding to the slot. Figure 5.2 shows the description of "U0-WALL-HEATING" and "U0-WALL-HEATING-DEMON" schemas consisting of the demon name and function name respectively. In this example, the demon function name is "find-u0-wall-heating" whose definition is shown in Figure 5.3. This function initializes some variables and then invokes appropriate functions

```
{{ U0-WALL-HEATING
            IS-A:       SLOT
            DEMON:      U0-WALL-HEATING-DEMON }}


{{ U0-WALL-HEATING-DEMON
            IS-A:       DEMON
            ACCESS:     GET-VAL
            WHEN:       AFTER
            EFFECT:     ALTER-VALUE
            ACTION:     FIND-U0-WALL-HEATING }}
```

Figure 5.2 CRL Print Forms of the Schemas Defining a Demon

```
(defun find-u0-wall-heating (schema slot demon
                              accessor access-values context)
 (setq type-of-building (get-value 'design-context 'type-of-building)
 (setq design-degree-days (get-value 'design-context
                                      'design-degree-days)
 (cond       ((equal type-of-building 1)
              (u0-walls-a1-heating design-degree-days))
             ((equal type-of-building 2)
              (u0-walls-a2-heating design-degree-days))
             ((equal type-of-building 3)
              (u0-walls-b1-heating design-degree-days))
             ((equal type-of-building 4)
              (u0-walls-b2-heating design-degree-days))
             ((t nil)) ) )
```

Figure 5.3: Example of a Demon Function

depending on the building type. If the building type belongs to the category 2, then the function "U0-B-BUILDING" shown in Figure 4.2 will be used. There are many such demons and functions for data retrieval, ensuring the dynamic updating of the design context for rapid design revisions.

The design alternative generator module has initialization routines and procedures for identifying feasible combination of envelope components. The initialization routines obtain the values for all the slots in the "design-context" schema and assign them to temporary variables used in the generation process. Design alternatives for building envelope are assembled by considering each component individually for its suitability and then combining them to meet the overall performance requirements. The alternative generator has a fixed top-level agenda and goes through a sequence of tasks described earlier in Figure 4.15, with constraint checking at various levels. This is implemented in commonLISP functions which perform a top-down and generate-test type of search. The information on material properties and construction types present in the knowledge base are used in the constraint checking process.

Once the alternative generation is complete, the alternatives are ranked based on their relative performance for each attribute. The ranking and selection processes are also implemented as commonLISP functions. These functions can handle any number of attributes for evaluation as long as they are defined with appropriate values representing their performance. Hence extending the BEADS knowledge base to address more performance attributes would require the addition of more information to the knowledge base and the inferencing modules would remain the same.

5.2.3 User-Interface

The end-user of the system is typically provided with prompts for input and results are displayed interactively with a series of questions and answers. Besides this sequential consultation process, the user may query the knowledge base for particular materials, construction types, feasible alternative descriptions, design revisions and re-evaluation. These tasks are highly individualistic during a consultation process and hence must be accessible without any restriction. In order to address the issues of designer freedom and easy access, DQL has been developed.

User-interface utilities perform type checking, input validations and prompting the user. Figure 5.4 shows a CommonLISP function "obtain-building-type" to interact with the user and obtain the building type under consideration. This user input is checked for legality and returned to the appropriate slot in the "DESIGN-CONTEXT" schema. Functional programming techniques and recursive feature are used in developing these functions. There are many such functions corresponding to all the user input parameters to define the design context and to rank the alternatives. These functions are also accessed by the DQL during design revisions and re-evaluations.

DQL is a command language processor which recognizes a finite number of keywords and transforms them to function calls or agenda items. A typical DQL query consists of an action keyword, object name, qualifiers and attributes if necessary, and a terminator symbol "*". Examples of DQL query statements are shown in Figure 5.5 and Appendix B lists the DQL keywords. Each query is parsed to determine the action to be taken. Each action may correspond to a simple function call for modifying the value of a design parameter or may be a search query requiring the creation of an agenda item. These are processed by the user-interface utilities or the inference mechanism or a combination of both. Currently DQL enables the following operations:

(i) Invoke the alternative generation and evaluation mechanisms independently. This is required for redesign and re-evaluation, once an initial design is completed.

(ii) Modify the design parameters individually at random. This allows the user to examine the influence of design parameters and to alter the design context.

(iii) Explore the knowledge base for alternatives and component descriptions with specified performance for attributes such as thickness, R-value and cost. This is necessary to simplify the search for schemas satisfying the performance constraints.

(iv) Add, display and delete descriptions in the knowledge base for building materials and construction types. These are required for knowledge base maintenance by the user who may not be familiar with the KNOWLEDGECRAFT development environment.

```
(defun  obtain-type-of-building  ( )
        (terpri)  (terpri)
        (princ  "Select the type of building  ")  (terpri)
        (princ  "     1.  Detached one or two-family dwelling")  (terpri)
        (princ  "     2.  Residential building - 3 stories or less")  (terpri)
        (princ  "     3.  Other buildings - 3 stories or less")  (terpri)
        (princ  "     4.  Other buildings - more than 3 stories")  (terpri)
        (princ  "Enter your selection <1-4>:   ")
        (setq  type-of-building  (read))
        (cond  ((and  (numberp  type-of-building)
                      (member  type-of-building  '(1  2  3  4)))
                 type-of-building)
               (t  (princ  "Wrong selection.  Please try again.")  (terpri)
                   (obtain-type-of-building)) ) )
```

Figure 5.4: Example of a User-Interface Function

```
DQL>  GENERATE  DESIGN  ALTERNATIVES  *
DQL>  EVALUATE  DESIGN ALTERNATIVES  *
DQL>  MODIFY  BUILDING  LOCATION  *
DQL>  FIND  A  FEASIBLE  ALTERNATIVE WITH  COST  <=  300000  *
DQL>  FIND  A  FEASIBLE  WALL  TYPE  WITH  R-VALUE > 2.0  *
DQL>  DISPLAY  LIST  OF  FEASIBLE ROOF TYPES  *
DQL>  FIND  A  ROOF  TYPE  WITH  THICKNESS  <=  200  *
DQL>  DISPLAY  DETAILS  OF  A  BASIC  WALL  TYPE  *
DQL>  ADD  A  ROOF  TYPE  *
DQL>  DELETE  A  GLAZING  TYPE  *
```

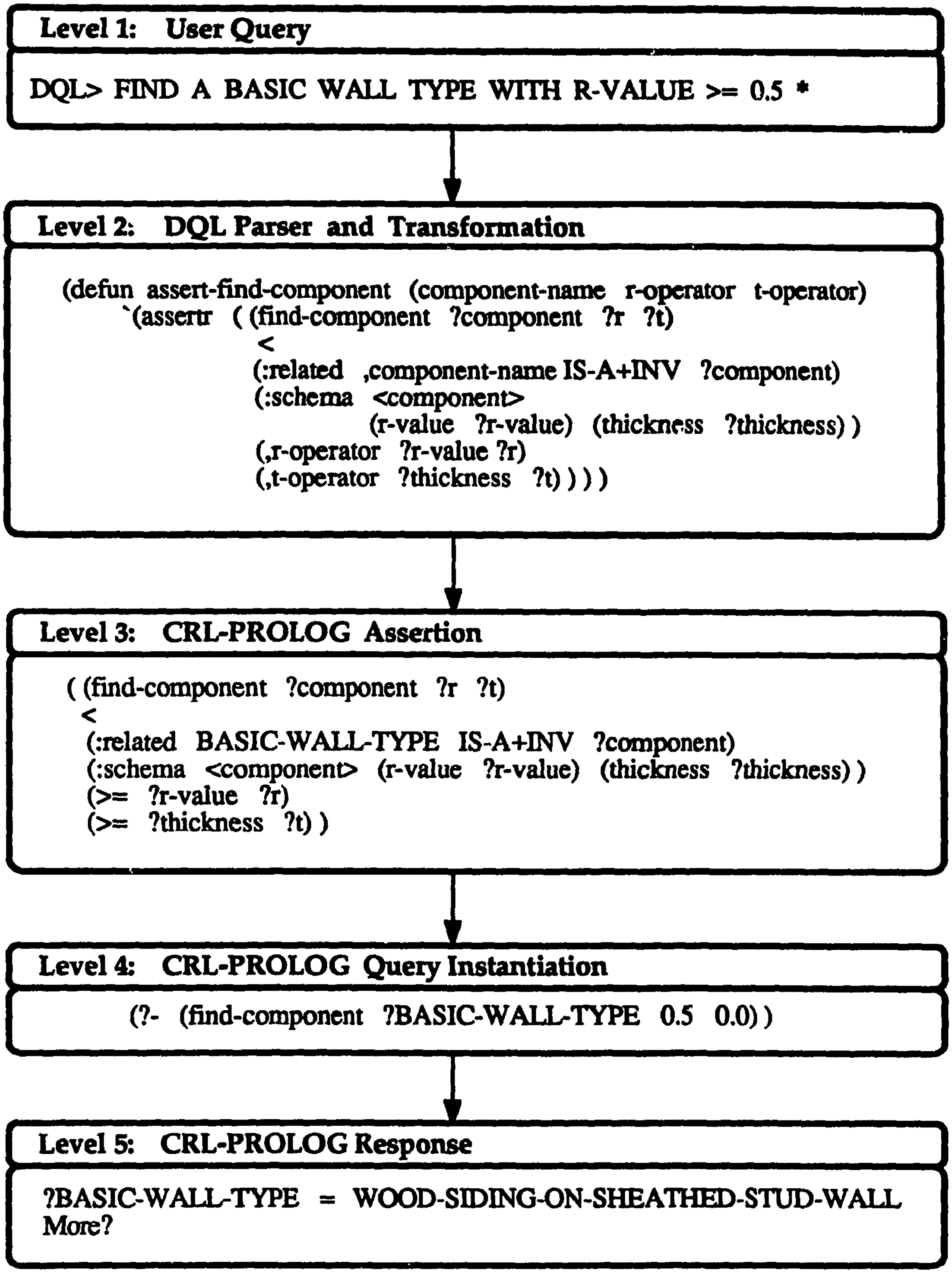
Figure 5.5: Example of DQL Query Statements

The DQL processor is implemented using CommonLISP functions built on top of KNOWLEDGECRAFT utilities. Most of the redesign and knowledge base maintenance features are directly related to the manipulation of schema descriptions and can be easily accomplished by transforming the queries to be function calls. But queries for searching the knowledge base are complicated due to the conditions and attributes which may vary between queries. DQL makes use of the CRL-PROLOG features available in KNOWLEDGECRAFT for performing the search. Each search query is translated to be a PROLOG axiom and asserted before the query is processed. Figure 5.6 shows the five levels of transformation that a search query goes through to obtain the desired result. The DQL query is first parsed to determine the action and then translated to a PROLOG assertion. The translation of DQL query to CRL-PROLOG is dynamic in nature and the predicates with target values are directly mapped from one form to the other. This assertion is added to the knowledge base. Then the CRL-PROLOG version of the query is instantiated to obtain the appropriate schema name.

5.2.4 Software Metrics

BEADS prototype development progressed in three phases. The first phase resulted in a preliminary prototype addressing only few performance attributes and provided the framework of knowledge representation and inference mechanisms. This preliminary prototype demonstrated the feasibility and enabled the identification of other essential performance attributes and knowledge required for improving the design alternative generation process. Interviews with architects and designers were conducted to review the knowledge base information and enhancements were made. Thus the second phase produced an improved version of the prototype knowledge base. The implementation of DQL to address the issues of designer freedom and knowledge base maintenance was completed in the final phase of implementation.

The knowledge base consists of about 110 schemas in total with varying number of attributes. For example, schemas of city names and weather data contain about 80 slots whereas the schemas of building materials and construction types have less than 10 slots. All these schemas are stored in a binary database format used by KNOWLEDGECRAFT and are automatically loaded in the environment before a consultation begins. There are about 80 CommonLISP functions corresponding to demons, inference mechanisms, user-interface and DQL utilities of the BEADS prototype.



```
DQL> FIND A BASIC WALL TYPE WITH R-VALUE >= 0.5 *
```

```
(defun assert-find-component (component-name r-operator t-operator)
     `(assertr ( (find-component ?component ?r ?t)
                 <
                 (:related ,component-name IS-A+INV ?component)
                 (:schema <component>
                          (r-value ?r-value) (thickness ?thickness) )
                 (,r-operator ?r-value ?r)
                 (,t-operator ?thickness ?t) ) ) )
```

```
( (find-component ?component ?r ?t)
  <
  (:related BASIC-WALL-TYPE IS-A+INV ?component)
  (:schema <component> (r-value ?r-value) (thickness ?thickness) )
  (>= ?r-value ?r)
  (>= ?thickness ?t) )
```

```
(?- (find-component ?BASIC-WALL-TYPE 0.5 0.0) )
```

```
?BASIC-WALL-TYPE = WOOD-SIDING-ON-SHEATHED-STUD-WALL
More?
```

Figure 5.6: Sequence of DQL Processing

Test problems have shown that the system can handle up to 1200 design alternative descriptions which may be added during a consultation. The execution time for generation of design alternatives range between 2 to 15 minutes for residential buildings and more than two hours for commercial buildings. The execution time for ranking and selection mechanisms is highly dependent on the number of alternatives and performance attributes considered in the process.

The preliminary prototype version of BEADS was also implemented in GOLDWORKS on the IBM PC. The knowledge representation and knowledge base information remained the same but the inferencing mechanisms were implemented in production rule forms. The execution time in GOLDWORKS-PC was much higher and in the order of 6 to 10 times slower than the KNOWLEDGECRAFT-VAX implementation.

5.3 Test Problems

Two design proposals were analyzed using the BEADS prototype to examine the validity and usefulness of the knowledge base. These design proposals were prepared during the preliminary stages of the respective building projects and were done by architects. Though there is sufficient information available for the envelope components, the basis for design and selection is neither documented nor available. The validation process is further complicated by the fact that the knowledge and experience of the designer are much broader in scope and specific to the Canadian construction practice when compared to that of the prototype knowledge base which is rather limited and derived from an American design handbook.

There are many other aspects of the process which are not yet exactly mapped into the BEADS consultation for comparison. Hence the test problems are carried out specifying no priorities on performance attributes but by specifying preferences on construction types which may result in designs comparable to the original ones. Information related to the cross section details for wall and roof systems are used in comparing the results. Missing data in interpreting the original design are obtained from the same sources as used for developing the BEADS knowledge base. The basis for validation essentially is the following performance attributes: thermal resistance, cost and thickness of wall, roof and glazings, and the overall energy consumption of the envelope assembly.

5.3.1 Residential Building

The first test problem is a three storey condominium building in Montreal. The preliminary design for this project was completed in 1985. The floor plan is trapezoidal as shown in Figure 5.7. Each floor consists of two double bedroom units, one triple bedroom unit and a single bedroom unit. This is a wood frame building with brick walls. Figure 5.8 shows an elevation of the building.

The BEADS consultation showing the area of envelope components and preferences specified in the design process are presented in Figure 5.9. There are 130 alternatives generated and all of them are found feasible, thus demonstrating the effectiveness of the search process. There are three basic wall types and five basic roof types which are feasible for this building. These basic construction types combined with different types of insulations and glazings result in the 130 feasible alternatives. During selection, preference on basic construction types are specified with a view to obtaining comparable solutions. Hence the BEADS suggested alternative consists of similar material compositions and exterior finish as that of the original design.

A graphical illustration of BEADS suggestions and original design details for external wall and roof assemblies are shown in Figure 5.10 and 5.11 respectively. Table 5.1 presents a comparison of performance attributes under consideration. The overall thickness of wall suggested by BEADS is greater than that of the original design. This is due to the fact that the stud space is filled with loose-fill insulation in the original design whereas BEADS provides a thicker wall insulation as a separate layer. It should be noted that loose-fill insulation performance deteriorates with time. Hence the energy consumption estimated with the present wall cross section details may not be valid for the long term performance of the original design. The R-value and energy consumption of the original design are very conservative and complies with the ASHRAE requirements. Though the thermal resistance of BEADS suggested roof system is less than that of the ASHRAE requirement, the overall energy budget of the envelope meets the standards requirements along with a reduction of about 20% of the material cost. The trade-off here is in selecting the optimum combination of envelope components rather than reducing the size of HVAC equipment.

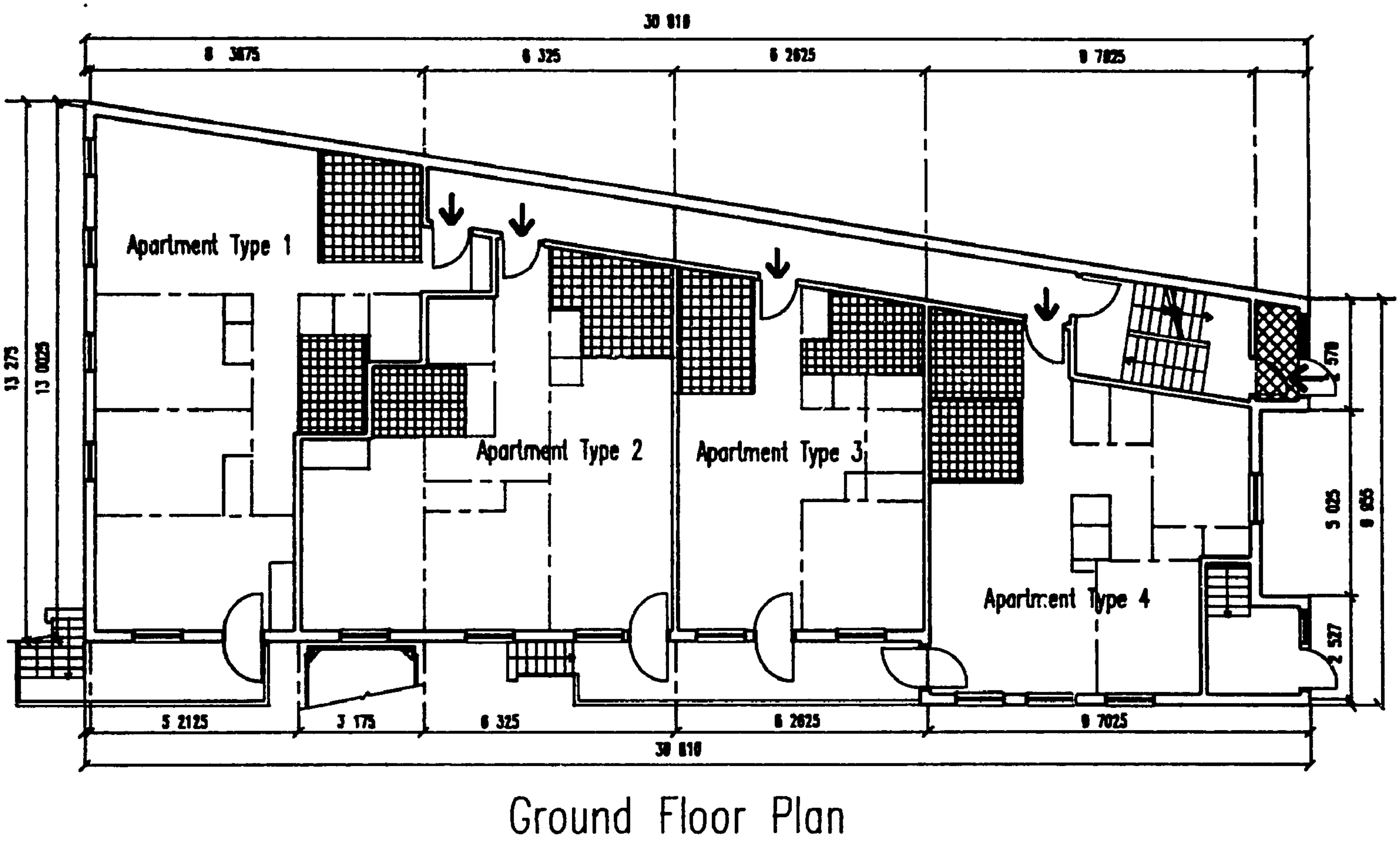
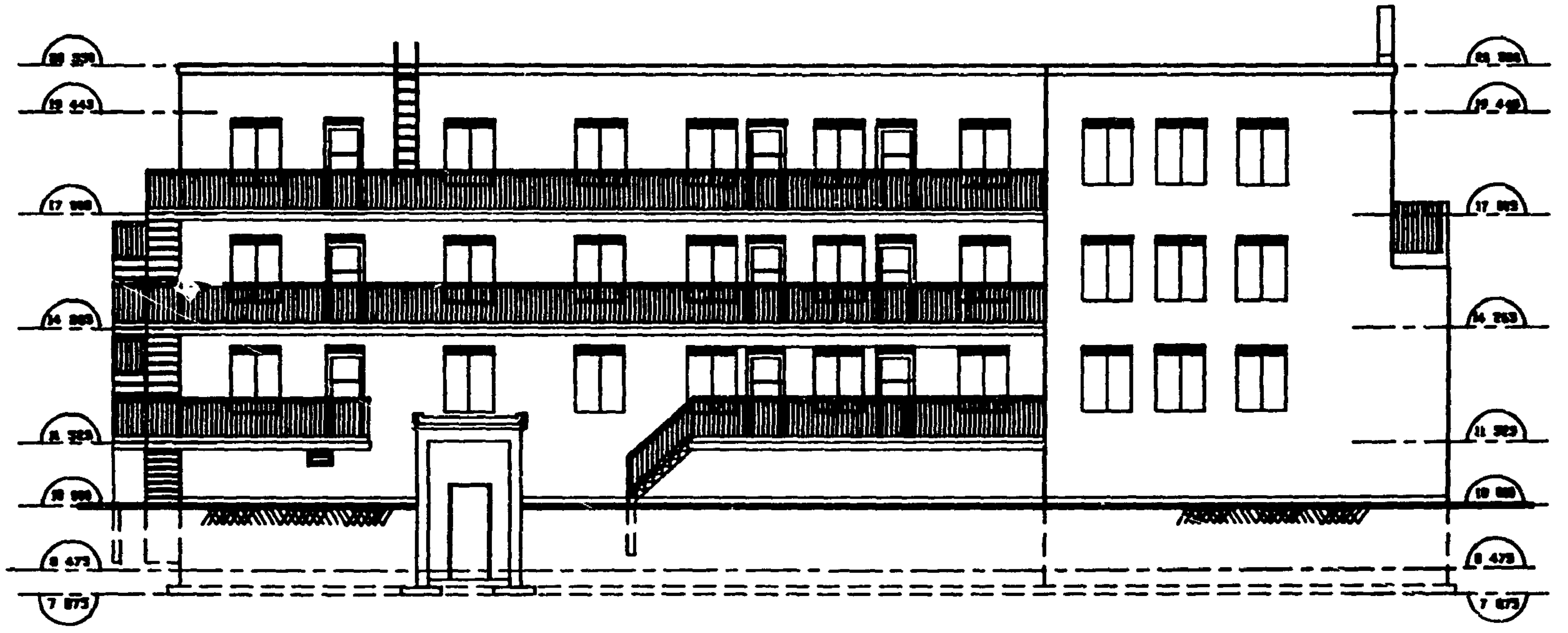


Figure 5.7: Floor Plan - Residential Building



South Elevation

Figure 5.8: Elevation - Residential Building

```
*** Welcome to BEAD Design Query Language Processor ***
You should terminate your query with a * or a ?
Type HELP at DQL prompt, if you need assistance

DQL> consult *

Give me the location of the proposed building: montreal

Select the type of building
        1. Detached one- or two-family dwelling
        2. Residential building - 3 stories or less
        3. Other buildings - 3 stories or less
        4. Other buildings - more than 3 stories

Enter your selection <1-4>: 2

Specify the structure type
          1. Wood frame
          2. Steel frame
          3. Concrete frame
          4. Composite frame
Enter your selection <1-4>: 1

Specify the roof shape:
          1. Flat roof
          2. Pitch > 3/12
          3. Pitch <= 3/12

Enter your selection <1-3>: 1

What is the total area of EXTERNAL-WALL (in sq.m)? 813

What is the total area of ROOF (in sq.m)? 111

What is the total area of FENESTRATION (in sq.m)? 116

What is the maximum permissible wall thickness (in mm)? 350

Starting to generate design alternatives. Please wait ...
```

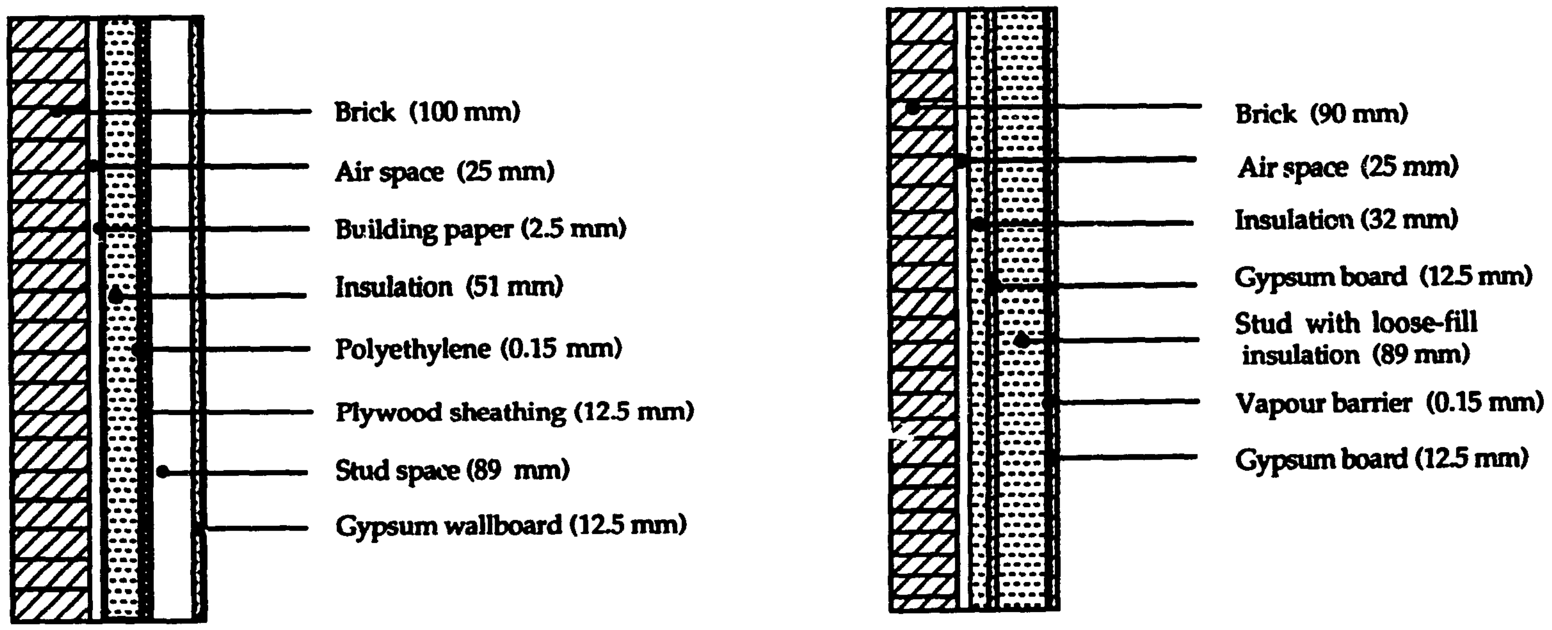
Figure 5.9: BEADS Consultation for Residential Building

```
Considering alternative no: 1
          WOOD-SIDING-ON-SHEATHED-STUD-WALL
          (GLASS-FIBREBOARD 51 14.6)
          SHEET-METAL-ON-PURLIN
          (RIGID-GLASS-FIBRE-RI 100 22.75)
          TRIPLE-GLAZING
Successful alternative no. 1

...
...

Considering alternative no: 130
          BRICK-ON-SHEATHED-STUD-WALL
          (GLASS-FIBREBOARD 51 14.6)
          BUILT-UP-OR-SINGLE-PLY-ON-WOOD-DECK
          (PHENOLIC-FOAM-BOARD 25 5.25)
          DOUBLE-GLAZING
Successful alternative no. 130


Alternative generation completed.
No. of alternatives generated are ... 130
No. of feasible alternatives are ... 130


The following basic wall types are feasible:
     1. BRICK-ON-SHEATHED-STUD-WALL
     2. STUCCO-ON-SHEATHED-STUD-WALL
     3. WOOD-SIDING-ON-SHEATHED-STUD-WALL

Enter your choice <(1 2 . . ) or ALL>: (1)

The following basic roof types are feasible:
     1. BUILT-UP-OR-SINGLE-PLY-ON-WOOD-DECK
     2. BUILT-UP-OR-SINGLE-PLY-ON-GYPSUM-DECK
     3. METAL-PANEL-ON-SHEATHED-RAFTER
     4. SHEET-METAL-ON-WOOD-DECK
     5. SHEET-METAL-ON-PURLIN

Enter your choice <(1 2 . . ) or ALL>: (1 2)
Please wait ...
```

Figure 5.9: BEADS Consultation for Residential Building (cont'd)

```
The following performance attributes can be
considered in the selection process:
    1. ENERGY-CONSUMPTION
    2. MATERIAL-COST
    3. TOTAL-THICKNESS-OF-WALL
    4. TOTAL-THICKNESS-OF-ROOF

Do you want to specify any preference among the
above performance attributes <Y/N>? n

The details of chosen alternative are:

  Basic wall type:          BRICK-ON-SHEATHED-STUD-WALL
  Wall insulation type:     GLASS-FIBREBOARD
  Basic roof type:          BUILT-UP-OR-SINGLE-PLY-ON-GYPSUM-DECK
  Roof insulation type:     PHENOLIC-FOAM-BOARD
  Glazing type:             DOUBLE-GLAZING

  Total wall thickness (mm):           292.65
  Wall insulation thickness (mm):      51
  Total roof thickness (mm):           345.15
  Roof insulation thickness (mm):      25

  Total material cost ($):        120422.6
  Utility values:                 (75.91 25.0 25.0 25.0 0.91)

Consultation completed successfully !!


DQL> exit *
```

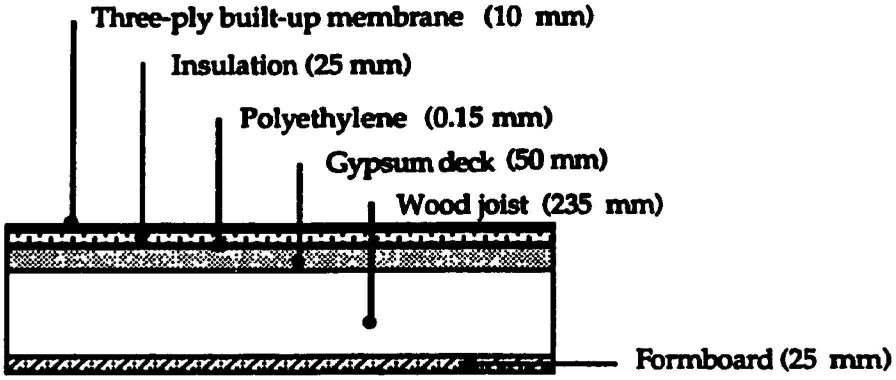
Figure 5.9: BEADS Consultation for Residential Building (cont'd)



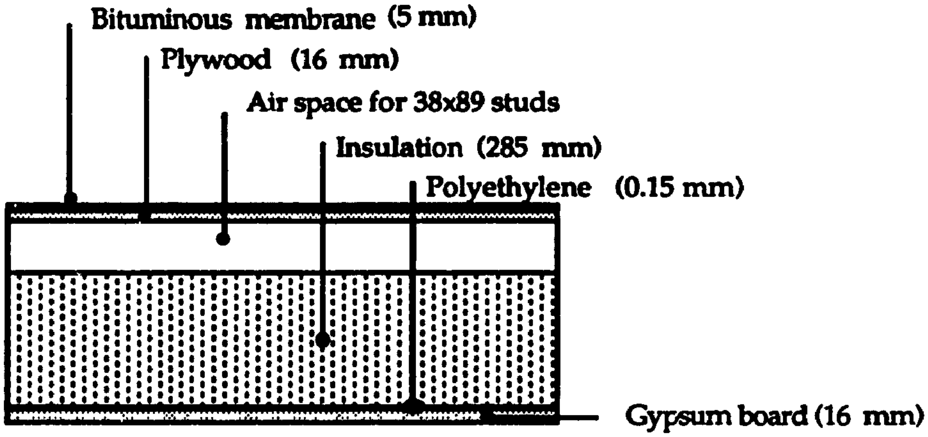
(i) BEADS suggested details

(ii) Original Design Details

Figure 5.10: Comparison of Wall Cross Section - Residential Building



(i) BEADS suggested details



(ii) Original Design Detail

Figure 5.11: Comparison of Roof Cross Section - Residential Building

Performance Attribute	BEADS Suggestions	Original Design	ASHRAE Requirements
Wall thickness (mm)	293	281	---
Roof thickness (mm)	345	411	---
R-value of wall (m^2 °C/W)	1.84	3.02	1.04
R-value of roof (m^2 °C/W)	2.05	5.88	5.34
Material Cost ($)	120 420	146 450	---
Energy consumption(W/°C)	862	678	1113

Table 5.1: Comparison of Performance Attributes - Residential Building

5.3.2 Industrial Building

The second test problem corresponds to a single storey industrial building with an attached office space. The design proposal for this project was completed in 1988 and this building is also located in Montreal. A site plan of this building is shown in Figure 5.12. The structural system consists of steel frames and steel deck. The area of glazings is less than 10 % of the external wall area. Figure 5.13 shows a perspective view of the building.

The BEADS consultation showing the details of envelope area and preferences for selecting the components are presented in Figure 5.14. There are 637 alternatives generated and 547 are identified as feasible ones. The relatively large number of alternatives generated in this case are due to the less stringent energy requirements for industrial buildings. As a consequence, many alternative combinations result in successful compliance to the ASHRAE requirements. BEADS generated solutions are once again narrowed down by specifying the choice of basic construction types.

The cross section details of external wall and roof of BEADS suggested and original design details are shown in Figures 5.15 and 5.16 respectively. Table 5.2 presents the comparison of performance. It can be observed that the overall thickness of BEADS suggested envelope components are smaller than that of the original ones and the individual R-values are considerably lower but sufficient enough to meet the overall requirements of the standard. Both designs satisfy the energy budget requirements and the original design is very conservative with higher R-values than what is actually required by the standard. The BEADS suggested alternative is 30% cheaper in material cost, though with a significant increase in energy consumption.

5.3.3. Discussion

It can be seen that the BEADS suggested envelope assembly is comparable in principle, though significantly different from the original design in terms of materials used and performance of individual components. The comparison of results reveals the lack of information in the knowledge base for considering the trade off in energy consumption to the material cost and its effect on envelope components. Also the number of construction types and building materials presently available in the system are not

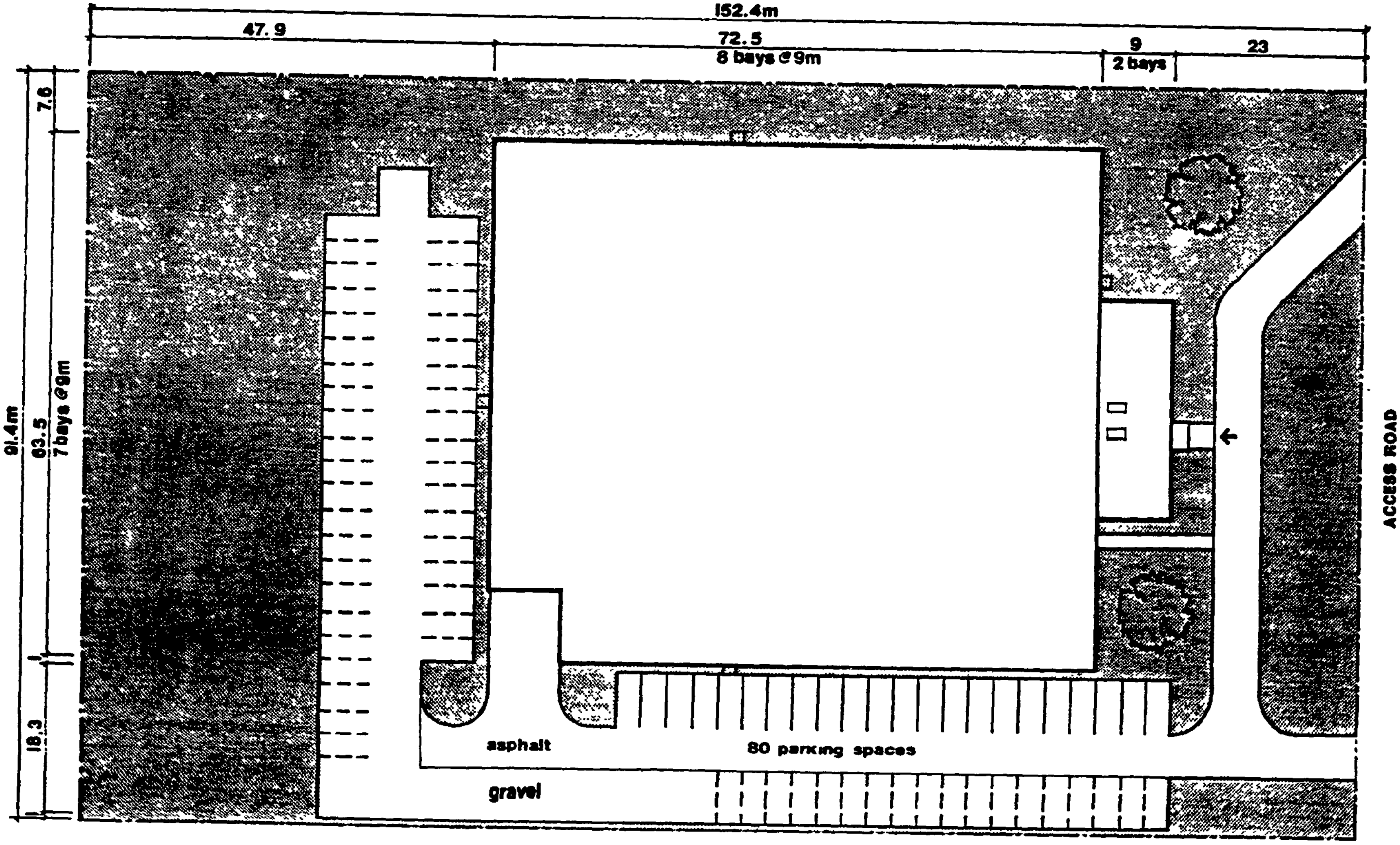


Figure 5.12: Site Plan - Industrial Building

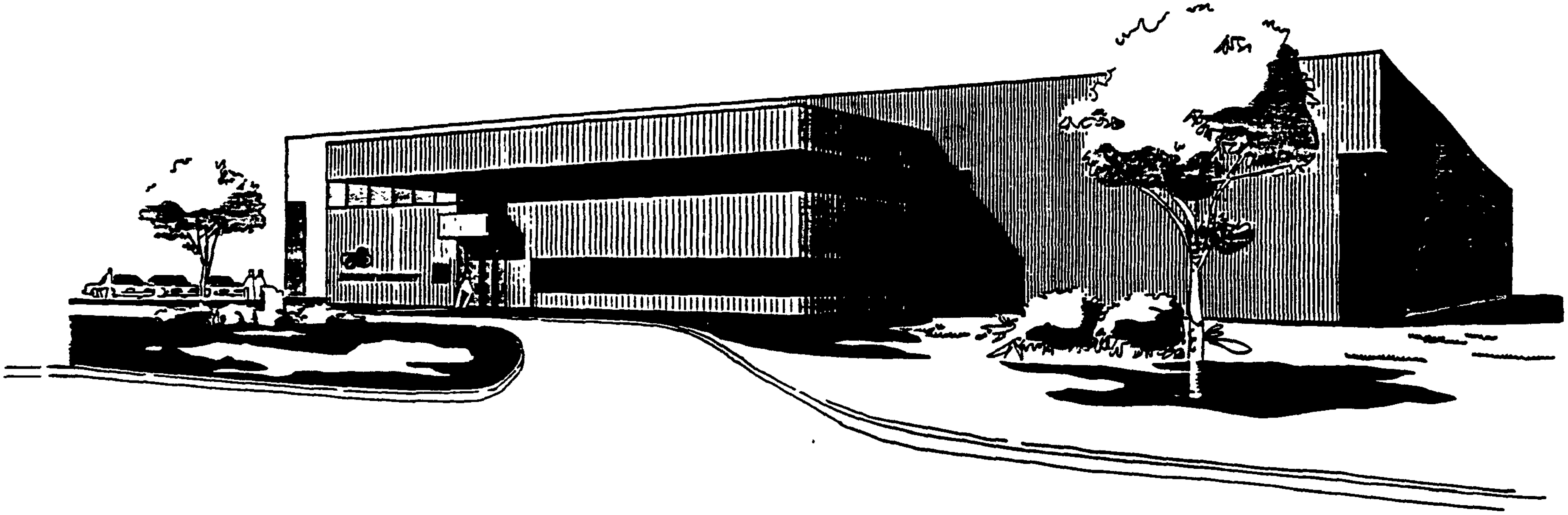
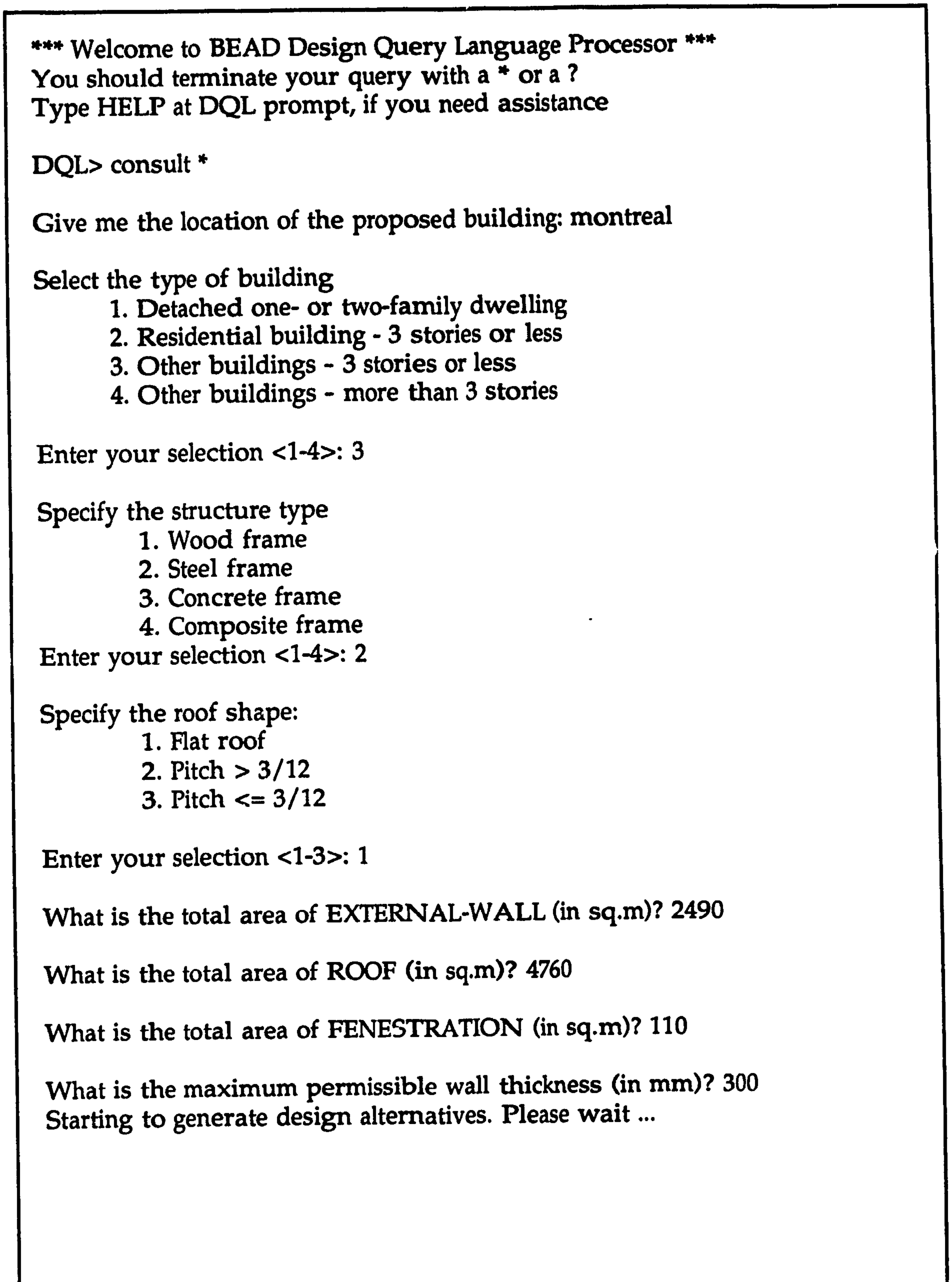


Figure 5.13: Perspective View - Industrial Building



```
*** Welcome to BEAD Design Query Language Processor ***
You should terminate your query with a * or a ?
Type HELP at DQL prompt, if you need assistance

DQL> consult *

Give me the location of the proposed building: montreal

Select the type of building
        1. Detached one- or two-family dwelling
        2. Residential building - 3 stories or less
        3. Other buildings - 3 stories or less
        4. Other buildings - more than 3 stories

Enter your selection <1-4>: 3

Specify the structure type
          1. Wood frame
          2. Steel frame
          3. Concrete frame
          4. Composite frame
Enter your selection <1-4>: 2

Specify the roof shape:
          1. Flat roof
          2. Pitch > 3/12
          3. Pitch <= 3/12

Enter your selection <1-3>: 1

What is the total area of EXTERNAL-WALL (in sq.m)? 2490

What is the total area of ROOF (in sq.m)? 4760

What is the total area of FENESTRATION (in sq.m)? 110

What is the maximum permissible wall thickness (in mm)? 300
Starting to generate design alternatives. Please wait ...
```

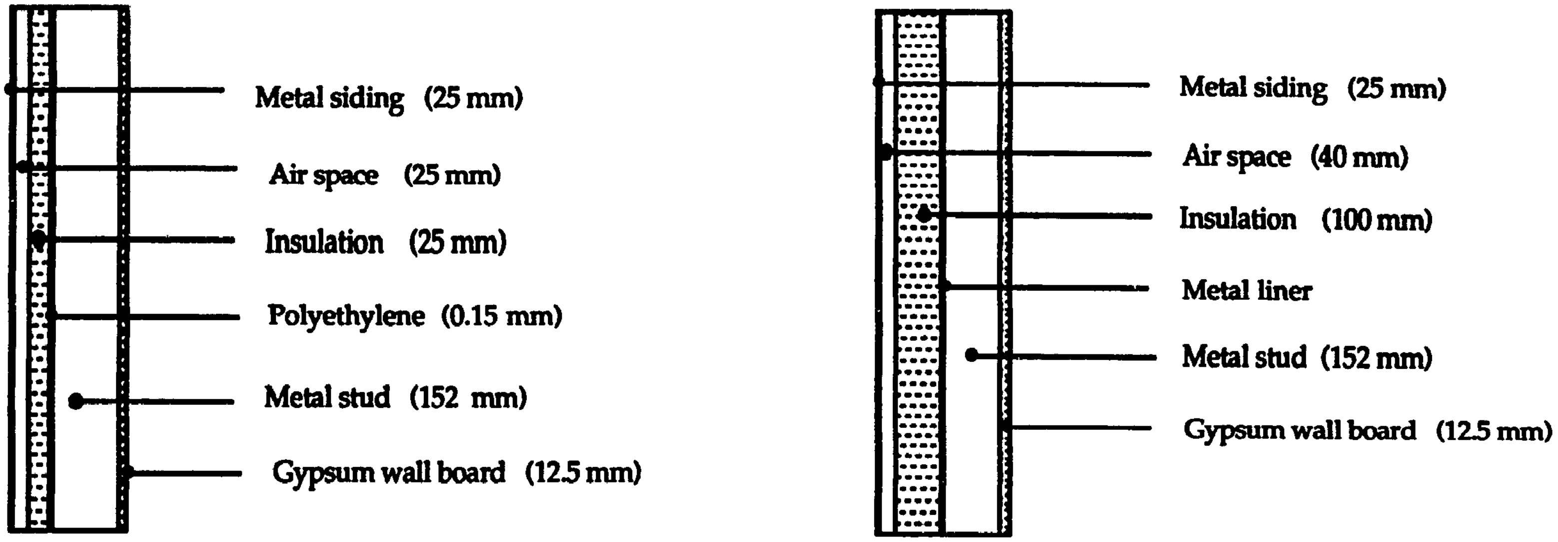
Figure 5.14: BEADS Consultation for Industrial Building

```
Considering alternative no: 1
          STUCCO-ON-CONCRETE-BLOCK-WALL
          (ISOCYANURATE-OR-RIGID-URETHANE-BOARD 38 17.25)
          SHEET-METAL-ON-PURLIN
          (RIGID-GLASS-FIBRE-RI 100 22.75)
          TRIPLE-GLAZING
Successful alternative no. 1

...

Considering alternative no: 636
          STONE-PANEL-ON-STEEL-FRAME-WALL
          (EXPANDED-POLYSTYRENE-TYPE-2 25 10.0)
          BUILT-UP-OR-SINGLE-PLY-ON-METAL-DECK
          (PHENOLIC-FOAM-BOARD 25 5.25)
          DOUBLE-GLAZING
Successful alternative no. 547

Alternative generation completed.
No. of alternatives generated are ... 637
No. of feasible alternatives are ... 547

The following basic wall types are feasible:
    1. STONE-PANEL-ON-STEEL-FRAME-WALL
    2. CONCRETE-BLOCK-WALL
    3. METAL-SIDING-ON-CONCRETE-BLOCK-WALL
    4. METAL-SIDING-ON-STEEL-GIRT-WALL
    5. STUCCO-ON-SHEATHED-STUD-WALL
    6. STUCCO-ON-CONCRETE-BLOCK-WALL
Enter your choice <(1 2 . . ) or ALL>: (4)

The following basic roof types are feasible:
    1. BUILT-UP-OR-SINGLE-PLY-ON-METAL-DECK
    2. BUILT-UP-OR-SINGLE-PLY-ON-PRECAST-CONCRETE-SLAB
    3. BUILT-UP-OR-SINGLE-PLY-ON-WOOD-DECK
    4. BUILT-UP-OR-SINGLE-PLY-ON-GYPSUM-DECK
    5. METAL-PANEL-ON-SHEATHED-RAFTER
    6. SHEET-METAL-ON-WOOD-DECK
    7. SHEET-METAL-ON-PURLIN

Enter your choice <(1 2 . . ) or ALL>: (1)
```

Figure 5.14: BEADS Consultation for Industrial Building (cont'd)

```
The following performance attributes can be
considered in the selection process:
    1. ENERGY-CONSUMPTION
    2. MATERIAL-COST
    3. TOTAL-THICKNESS-OF-WALL
    4. TOTAL-THICKNESS-OF-ROOF

Do you want to specify any preference among the
above performance attributes <Y/N>? n

The details of chosen alternative are:

  Basic wall type:          METAL-SIDING-ON-STEEL-GIRT-WALL
  Wall insulation type:     EXPANDED-POLYSTYRENE-TYPE2
  Basic roof type:          BUILT-UP-OR-SINGLE-PLY-ON-METAL-DECK
  Roof insulation type:     PHENOLIC-FOAM-BOARD
  Glazing type:             DOUBLE-GLAZING

  Total wall thickness (mm):            239.65
  Wall insulation thickness (mm):       25.0
  Total roof thickness (mm):            154.50
  Roof insulation thickness (mm):       38.0

  Total material cost ($):        1235800.0
  Utility values:                 (72.84 20.71 25.0 25.0 2.13)

Consultation completed successfully !!
```

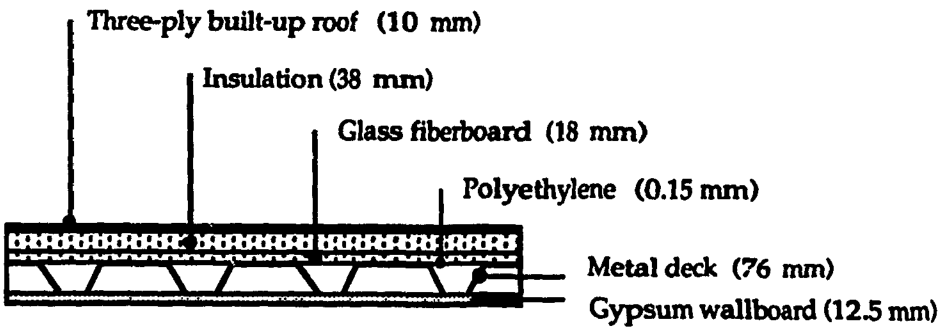
Figure 5.14: BEADS Consultation for Industrial Building (cont'd)



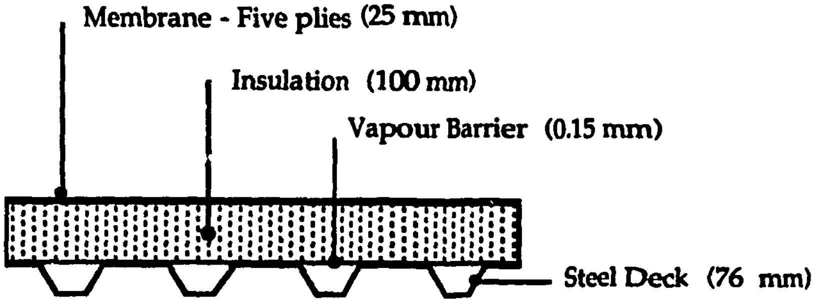
(i) BEADS suggested details

(ii) Original Design Details

Figure 5.15: Comparison of Wall Cross Section - Industrial Building



(i) BEADS suggested details



(ii) Original Design Detail

Figure 5.16: Comparison of Roof Cross Section - Industrial Building

Performance Attribute	BEADS Suggestions	Original Design	ASHRAE Requirements
Wall thickness (mm)	240	313	---
Roof thickness (mm)	155	200	---
R-value of wall (m^2 °C/W)	1.27	2.69	0.83
R-value of roof (m^2 °C/W)	2.07	3.08	2.93
Material Cost ($)	1 235 800	1 507 600	---
Energy consumption(W/°C)	4582	2830	4914

Table 5.2: Comparison of Performance Attributes - Industrial Building

extensive enough to cover the broad range of products and techniques used in design practice. The structural system though considered to a certain extent, is not sufficient. As discussed earlier, there are many other factors contributing to the differences between BEADS suggestions and original designs. Nevertheless, the test problems illustrate the suitability of BEADS methodology in exploring the alternatives at the preliminary design stage. Further the suggested solutions are found to be practically viable ones based on the information available in the knowledge base. The designers who examined these suggested solutions had no reservation against using these designs, but still said that they would prefer the ones already used in their practice.

5.4 Comments on the BEADS Prototype

In its various stages of development, the BEADS prototype was presented to architects, engineers and designers ([74], [75], [76], [77]). The response has been very positive, encouraging and sometimes critical. Many architects expressed an interest to use this system in practice if it were commercially available. They also desired some additional features with respect to graphical representation of envelope components, accessability to modify the knowledge base and develop their own. One of the prime reasons for their interest is to explore design alternatives and provide pre-tender estimate for the building envelope when there is little information available in the earliest stages of design. One architect pointed out that frequent revisions to envelope components arise from budget constraints where the developer requires alternative proposals within a very short period of time. BEADS will assist in this situation and ensure that the performance requirements are not violated. Also the knowledge base is viewed as a medium for knowledge transfer from experts to novices in the profession.

Critics of the system are skeptical about the applicability of BEADS knowledge base in its present form. Their main concern originates from the fact that only a few performance attributes are addressed in the prototype knowledge base. Though this is a valid concern, the current work has focused primarily on developing a framework for knowledge representation and now provides a platform for incorporating additional performance attributes readily. A commercial version of BEADS may be developed to consider more performance attributes and extend the knowledge base with additional construction types, building materials, portions of Codes and Standards as required.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

This thesis presents a systematic approach and framework for automating the building envelope design process. Knowledge-based system techniques are utilized to solve the problems of information handling and decision making for synthesis and evaluation of design alternatives. The BEADS prototype implementation demonstrates the feasibility and practicality of developing software tools to assist designers in making knowledgeable design decisions at the earliest stages when there is not enough information available. The major conclusions of the present investigation are summarized below:

(i) **Complexities in the Building Envelope Design Process:** Selection of materials and construction types for building envelope components is subject to many conflicting performance requirements depending on climatic conditions, building regulations and the objectives of the owner. The design process is further complicated by the need to consider a large number of alternative combinations of materials and construction types to achieve economy and successful performance. Building envelope design decisions are often made at the preliminary design stage when the schematic drawings are prepared. Lack of information and resources prohibit the designer from doing any detailed investigation of the decisions being made. This results in sub-optimal design solutions frequently leading to problems during construction and other performance deficiencies during the life cycle of the building. It is found that there is a lack of tools and techniques to address the building envelope design problems encountered at the preliminary design stage.

(ii) **Design Methodology:** The number of performance attributes and alternatives to be considered in the design process are so large that the designer cannot solve them manually. Systematic approaches and computer aided design models have been reviewed and the following stages in building envelope design are identified: (i) establishing the design context, (ii) generation of design alternatives and (iii) ranking and selection.

These three stages realistically represent the sequence of decisions made and allow the decomposition of the design process into independent tasks which interact to produce the final design. Exploratory search processes are sought to perform the generation of design alternatives.

(iii) **Knowledge-Based System Approach:** Automating the building envelope design process requires the integration of many different types of knowledge and information which is difficult to achieve using procedural programming techniques. It is shown that knowledge-based system techniques are best suited to represent heuristic, procedural and database information using versatile knowledge representation paradigms. A knowledge base containing the construction types, building materials and other information required in design can be developed and integrated with an inference mechanism for generation of design alternatives and to assist in decision making. Such an approach allows the seperation of knowledge base from the inference process thus providing a framework for incremental development and easy updating.

(iv) **Knowledge-Based System Development Tools:** Many commercially available development tools have been evaluated by implementing prototype design problems. This evaluation identifies the various aspects of selecting a development tool for engineering design applications. Building envelope components are essentially objects with certain functional and performance attributes which need to be represented in schemas. It is found that hybrid systems providing both schemas (frames) and rules are required to represent procedural and heuristic knowledge in an efficient manner. KNOWLEDGECRAFT is chosen as a viable tool for developing the knowledge base and inference mechanism required in building envelope design.

(v) **Knowledge Representation:** The prototype knowledge base demonstrates the integration of information from building code, performance standards, design manuals, material properties and cost data. A hierarchy of building envelope components is developed to semantically relate the materials and construction types. Functional and performance attributes for materials are represented as slots and values of generic materials and construction types schemas. These include both performance data and design heuristics. Material properties and construction type descriptions are obtained from technical literature. Design heuristics on compatibility between materials and suitability of using construction types in a particular design context are extracted by interviewing designers. It is shown that quantitative performance requirements can be established by interfacing to procedural programs to the attributes in the design context schema. The dynamic relationships between user requirements and performance attributes in the design context ensures the integrity and consistency of information for rapid design revisions.

(vi) **Design Alternative Generation:** A sequential process for generating building envelope design alternatives is utilized to systematically explore the possible combinations of materials and construction types. Constraint checks at material, component and system levels are performed to ensure the efficiency of the search process. The semantic relationships between materials and envelope components are resolved during this process. The prototype system demonstrates the use of this technique to determine the thickness and type of insulation material at the component level. By decomposing the design process, it is shown that multiple levels of "generate-test" can be implemented for constraint-based search.

(vii) **Ranking and Selection Processes:** The identification of feasible alternatives and their performances provide the basis for ranking and selection of alternatives. Designers often have preferences and priorities which are specific to each design context. The present study shows that simple decision methodologies are sufficient to compare the alternatives on a normalized scale. Utility values for each performance attribute of all the alternatives are calculated by relative ranking. Designer priorities on performance attributes are used to establish the weighted overall utility for the alternatives. Such an approach to ranking and selection is found suitable at the preliminary design stage when the precise prediction of performance for each alternative is neither possible nor necessary. Since the feasible alternatives are retained in the knowledge base, re-evaluation can be easily done.

(viii) **Design Query Language:** The need for designer freedom and knowledge base maintenance are critical to the practical value of knowledge-based design tools. The development of a design query language meets this requirement by providing a problem-oriented command language to query the knowledge base and to perform the design operations when necessary. Selection of individual materials, construction types or design alternatives with specific levels of performance are possible with this approach. Modification of particular design parameters for revising a design context, generation, ranking and selection can be done as desired. This design query language also serves as a knowledge acquisition interface for maitaining and updating the knowledge base without expecting the user to be familiar with the development environment.

(ix) **BEADS Prototype Implementation:** The feasibility of automating the building envelope design process is demonstrated by the BEADS prototype implementation. The prototype knowledge base represents the integration of various types of design information in a structured form. Domain dependent inference procedures have been developed for establishing the design context, generation of alternatives, and ranking and selection processes. The design query language and user-interface utilities provide designer freedom to exploit the knowledge base for rational decision making. This emphasizes the fact that BEADS is not a black box for design but a tool to enhance the designers abilities to experiment with design concepts.

(x) **Testing and Validation:** Two practical design problems have been solved using the prototype system and the results compared to original designs. There are some significant differences in material compositions between the original designs and BEADS suggestions, though the quantitative performances are comparable. Lack of sufficient information to establish the design context, limitations of the prototype knowledge base are primary reasons for the discrepencies. But these test problems in effect illustrate the applicability of BEADS approach to solve real world design problems. Architects and designers have expressed an interest in using such a tool if it were commercially developed and available.

6.2 Summary of Contributions

The present study contributes to a better understanding of the problems in the building envelope design process and the need to develop computer-aided design tools. These are addressed by developing a framework for systematic approach to design decision making. It is shown that knowledge-based system approach to building envelope design can greatly reduce the problems of information handling and decision making encountered at the preliminary design stage. Such an approach will assist the designer with a knowledge base to consider many design alternatives and multiple performance attributes when there is only limited information available for the design context. Though the prototype knowledge base accounts for only a few performance attributes, the knowledge representation methodology is versatile for incorporating additional attributes with little effort. Issues of designer freedom for exploration, integration of different types of information and search techniques for alternative generation are successfully realized with BEADS implementation.

The design methodology and knowledge-based system framework described in the present study can be adapted to other design domains where design descriptions need to be developed from functional and performance specifications. Formulation type design problems in general can be automated using the problem decomposition and multiple levels of "generate-test" paradigms as demonstrated for the generation of feasible design alternatives. In summary, the present study contributes to the research in developing tools and techniques for the preliminary design stage, specifically the selection of materials and construction types for building envelope components.

6.3 Recommendations for Further Research

The development of a systematic approach and design framework is only a first attempt towards automating the building envelope design process. There are many research issues remain to be investigated for improving the knowledge base, incorporating better design techniques at component level, implementing simplified analysis techniques for performance prediction and integrating with other building subsystems design. Some of the most important concerns identified during the course of the present study are described below:

(i) **Extending BEADS knowledge Base:** Energy efficiency, interstitial condensation in wall assemblies, structural and material compatibilities are the major performance attributes considered in implementing the prototype system. Other aspects such as structural strength, fire and acoustic resistances, air-barrier characteristics and durability need to be addressed in design. Extensive research must be done to identify the appropriate information for these attributes so that their quantitative performance requirements can be defined in the design context. In addition, the material properties data must be obtained and performance prediction for alternatives must also be developed. The knowledge base must also be updated with more building materials and construction types before it can be used in practice. The ASHRAE 90A Standard is being replaced with two new Standards 90.1 and 90.2 for commercial and residential buildings respectively. These must be incorporated in the BEADS knowledge base.

(ii) Graphic Interface to the Knowledge Base: Designers often prefer to examine a graphic representation of envelope components and their material composition. Such an interface to BEADS knowledge base needs to be developed for displaying the details of construction types and design alternatives. The data exchange formats between the knowledge base and graphics interface must be established before implementation. Once developed, this interface will enhance the designers ability to visualize the design details as to the number of layers, materials and thicknesses, etc. Prototype efforts on the graphic interface issues are already being pursued at the Centre for Building Studies.

(iii) Integration of Design and Diagnosis: There is abundant knowledge available on building envelope failure diagnosis and case studies. Systematic approach to building envelope diagnosis can be achieved using knowledge-based system techniques [78]. If such a knowledge base is developed in a generic form, then it can be used to check new design proposals for successful performance. A new knowledge representation framework is necessary to support both diagnosis and design with the same knowledge base. Though this is feasible as illustrated by Fazio and Gowri [79] for moisture induced problems, a lot more research needs to be done for addressing the many possible building envelope failures.

(iv) Alternative Generation at a Lower Level of Abstraction: The BEADS alternative generation is implemented at the component level in which a set of already available basic construction types are used with insulation as the only material to be identified in the design process. But all building materials can be defined with their functional attributes and material compatibility information which can then be used by generic construction type definitions for generating innovative combination of materials. It is clear that the generation process will be slower and hence more emphasis on heuristic information is required to generate meaningful solutions.

(v) Other Ranking and Selection Techniques: There are many optimization and decision making methodologies already available for selection of design alternatives. The application of these methodologies has not been successful because of the extensive amount of information and detail required to formulate the optimization problem and in understanding the mathematical complexities. BEADS utilizes a simple weighted utility value method for comparing the alternatives. But this may be enhanced with more sophisticated techniques by taking advantage of the performance prediction calculations made during alternative generation. More work is needed to identify if these are possible and if so the implementation techniques have to be developed as well. A suggestion here is to provide a number of ranking and selection techniques, allowing the user to select an appropriate one.

(vi) **Incorporating Detailed Simulation in Preliminary Design:** A recent study by Shaviv and Kalay [80] shows that procedural and heuristic methods can be combined by integrating simulation models with knowledge-based systems for all phases of energy conscious design. The possiblity of developing a similar technique for building envelope design must be investigated to take advantage of the available performance prediction and simulation techniques. This strategy will provide the designer with a knowledge base for making appropriate design assumptions required for detailed analysis, even at the preliminary design stage.

(vii) **Integrating BEADS with Other Building Subsystems Design:** Currently BEADS attempts to solve the building envelope design problem in isolation from other subsystems such as structure and HVAC. It is possible to develop independent preliminary design tools for each subsystem design. The interaction between the various subsystems is critical to the overall design of the building. Integrated building design issues seem to be the most important presently for improving the design process. Hence further research should be undertaken to examine the influence of design decisions and their effect on related subsystems.

REFERENCES

1. Fazio, P., "A Case Study in Typical Building Design - Highlighting the Need for Improved Integration of Design", Canadian Joul. of Civil Engineering, August 1990.

2. Building Envelope Failures: Causes and Remedial Measures, No.1: Walls, Alberta Building Envelope Council, Edmonton, Alberta, 1985.

3. Building Envelope Failures: Causes and Remedial Measures, No.2: Roofs, Alberta Building Envelope Council, Edmonton, Alberta, 1985.

4. Building Envelope Failures: Causes and Remedial Measures, No.3: Windows, Alberta Building Envelope Council, Edmonton, Alberta, 1985.

5. Mackinder, M., "The Selection and Specification of Building Materials and Components", Institute of Advanced Architectural Studies Research Paper No. 17, The University of York, York, U.K., 1980.

6. Kalay, Y. E., "Redefining the Role of Computers in Architecture: From Drafting / Modelling to Knowledge-Based Assistants", Computer Aided Design, Vol. 17, No. 7, pp. 319 - 328, 1985.

7. Logcher, R. D., "ICES STRUDL: An Integrated Approach to a Computer System for Structural Engineering", Emerging Methods in Environmental Design and Planning, ed. G. T. Moore, Cambridge, MA, pp. 79 - 91, 1970.

8. Building Research Board, "The 1984 Workshop on Advanced Technology for Building Design and Engineering", National Academy Press, Washington, D. C., 1985.

9. Dell' Isola, A. J., "Value Engineering in the Construction Industry", Van Nostrand Reinhold, New York, NY, 1982.

10. Corkan, J., "Check-list for Design to Overcome Building Envelope Failures", Building Envelope Failures: Causes and Remedial Measures, Seminar No. 1 - Walls, Alberta Building Envelope Council, Calgary, Alberta, pp. 87 - 103, March 1985.

11. Hutcheon, N.B., "Requirements for Exterior Walls", Canadian Building Digest No. 48, Division of Building Research, National Research Council of Canada, December 1963.

12. Elder, A.J. and Vandenberg, M., "AJ Handbook of Building Enclosure", The Architectural Press, London, 1974.

13. Design: The Building Envelope, Architects Handbook of Energy Practice, The American Institute of Architects, 1982.

14. Guide to energy Efficiency Masonry and Concrete Buildings, Masonry Council of Canada, Downsview, Ontario, April 1982.

15. Hutcheon, N. B. and Handegord, O. P., "Building Science for a Cold Climate", John Wiley and Sons, New York, NY, 1983.

16. Reid, E., "Understanding Buildings - A Multidisciplinary Approach", The MIT Press, Cambridge, MA, 1987.

17. Mattar, S. G., "Decision by Exclusion: On the Design of the Building Enclosure", D. Eng. Thesis at the Centre for Building Studies, Concordia University, Montreal, 1976.

18. Finger, S. and Dixon, J. R., "A Review of Research in Mechanical Engineering Design, Part I: Descriptive, Prescriptive and Computer-Based Models of Design Processes", Research in Engineering Design, Vol. 1, No. 1, pp. 51 - 68, 1989.

19. Willey, D. S., "Approaches to Computer-Aided Architectural Sketch Design", Computer Aided Design, Vol. 8, No. 3, pp. 181 - 186, 1976.

20. Willey, D. S., "Structure for Automated Architectural Sketch Design", Computer Aided Design, Vol. 10, No. 5, pp. 307 - 312, 1978.

21. Bedard, C. and Gowri, K., "Automating the Building Design Process with KBES", Joul. of Computing in Civil Engineering, American Society of Civil Engineers, Vol. 4, No. 2, pp. 69 - 83, 1990.

22. Manning, P., "A Strategy for Multiprofessional Design of Buildings", Canadian Journal of Civil Engineering, Vol. 11, No. 1, pp. 25 - 34, 1984.

23. Dieter, G. E., "Engineering Design: A Materials and Processing Approach", McGraw-Hill, New York, NY, 19[illegible]

24. Kalay. Y. E., Swerdloff, L. and Majowski, B., "Process and Knowledge in Design Computation", Journal of Architectural Education, Vol. 43, No. 2, 1990.

25. Gross, M., Ervin, S., Anderson, J. and Fleisher, A., "Designing with Constraints", Computability of Design, ed. Y. E. Kalay, John Wiley and Sons, New York, NY, pp.53 - 83, 1987.

26. Chan, W-T. and Paulson, B. C., "Exploratory Design Using Constraints", Artificial Intelligence in Engineering Design, Analysis and Manufacturing, Vol. 1, No. 1, pp. 59 - 71, 1987.

27. Radford, A. D. and Gero, J. S., "Design by Optimization in Architecture, Building and Construction", Van Nostrand Reinhold, New York, NY, 1988.

28. Arlani, G-A., "Optimum Building Design Decision Making", Ph. D. Thesis at the Centre for Building Studies, Concordia University, Montreal, 1986.

29. Murthy, N. S. H., "Multifunctional Building Enclosure Design", Ph. D. Thesis at the Department of Architectural Science, University of Sydney, Australia, 1983.

30. D'Cruz, N.A. and Radford, A.D., "A Multicriteria Model for Building Performance and Design", Building and Environment, Vol.22, No.3, pp 167-169, 1987.

31. Radford, A.D. and Gero, J.S., "On Optimization in Computer Aided Architectural Design", Building and Environment, Vol.15, pp. 73-80, 1980.

32. Gero, J.S., D'Cruz, N.A. and Radford, A.D., "Energy in Context: A Multicriteria Model for Building Design", Building and Environment, Vol.18, pp. 99-107, 1983.

33. Tham, K. W., Lee, H. S. and Gero, J. S., "Building Envelope Design using Design Prototypes", working paper at The Design Computing Unit, Dept. of Architectural and Design Science, University of Sydney, Australia, Feb. 1990.

34. Maher, M. L. and Fenves, S. J., "HI-RISE: An Expert System for the Preliminary Structural Design of High Rise Buildings", Knowledge Engineering in Computer Aided Design, International Federation of Information Processing (IFIP), Budapest, Hungary, 1984.

35. Gero, J. and Coyne, R. D., "Developments in Expert Systems for Design Synthesis", in Expert Systems in Civil Engineering, edited by C. N. Kostem and M. L. Maher, American Society of Civil Engineers, NY, 1986.

36. Rosenman, M. A. and Gero, J. S., "Design Codes as Expert Systems", Computer Aided Design, Vol. 17, No. 9, pp. 399 - 409, 1985.

37. Garrett, J. H. Jr. and Fenves, S. J., "A Knowledge-Based Standards Processor for Structural Component Design", Engineering with Computers, Vol. 2, pp. 219 - 238, 1987.

38. Sriram, D., "Knowledge-Based Approaches for Structural Design", Ph. D Thesis at the Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, 1986.

39. Fenves, S. J., Flemming, U., Hendrickson, C., Maher, M. L. and Schmitt, G., "Integrated Software Environment for Building Design and Construction", Computer-Aided Design, Vol. 22, No. 1, pp. 27 - 36, 1990.

40. Bourdeau, M., "Artificial Intelligence and Building CAD: Example in Handling Structured Objects by an Expert System to Simulate Design Tasks", Proc. of the European Conference on Architecture, Munich, April 1987.

41. Working with the Performance Approach in Building, CIB Report No. 64, International Council for Building Research Studies and Documentation, Rotterdam, Netherlands, 1982.

42. Rosen, H. J. and Bennett, P. M., "Construction Materials Evaluation and Selection - A Systematic Approach", John Wiley and Sons, New York, NY, 1979.

43. ASHRAE Standard 90A-1980 - Energy Consevation in New Building Design, American Society of Heating, Refrigerating and Air-Conditioning Engineers Inc., New York, NY, 1980.

44. Mackinder, M., "Design Decision Making in Architectural Practice", Institute of Advanced Architectural Studies Research Paper No. 19, The University of York, York, U.K., 1982.

45. Gero, J. S. and Coyne, R., "The Place of Expert Systems in Architecture", CAD 84, Butterworths, Guildford, pp. 529 - 546, 1984.

46. Sriram, D., Maher, M. L. and Fenves, S. J., "Knowledge-Based Expert Systems in Structural Design", Computers and Structures, Vol. 20, No. 1-3, pp. 1 - 9, 1985.

47. VAX OPS5 User's Guide and Reference Manual, Digital Equipment Corporation, Maynard, MA, 1988.

48. The OPS5+ Development System, Reference Manual, Artelligence Inc., 1986.

49. Watson, M., "ExperOPS5 User's Manual", SAIC and Expertelligence Inc., 1985.

50. Brownston, L., Farrell, R., Kant, E. and Martin, N., "Programming Expert Systems in OPS5 - An Introduction to Rule-Based Systems, Addison-Wesley, Reading, MA, 1985.

51. Canadian Standards Association, "Design of Reinforced Concrete Structures for Buildings", CAN3-A23.3-M84, Rexdale, Ontario, 1984.

52. Carnegie Groups Inc., "KNOWLEDGE CRAFT: Reference Manual", Pittsburg, PA, 1987.

53. Fazio, P., Zmeureanu, R. and Kowalski, A., "Select-HVAC: Knowledge-Based System as an Advisor to Configure HVAC Systems", Computer Aided Design, Vol. 21, No. 2, pp. 79 - 86, 1989.

54. Micro Data Base Systems Inc., "GURU: Reference Manual", Lafayette, IN, 1985.

55. Associate Committee on National Building Code of Canada, "National Building Code of Canada", National Research Council of Canada, Ottawa, Ontario, 1985.

56. Associate Committee on the National Building Code, Supplement to the National Building Code of Canada, National Research Council of Canada, Ottowa, Ontario, 1985.

57. Texas Instruments Inc., "Personal Consultant Plus: User's Guide and Reference Manual", Austin, TX, 1986.

58. Gold Hill Computers Inc., "GOLDWORKS: Expert System Users Guide", Cambridge, MA, 1987.

59. Technowledge Inc., "M.1 Reference Manual", Palo Alto, CA, 1985.

60. Billman, C. and Norman, W., "LEVEL5 Tutor: Study Guide", Information Builders Inc., Indialantic, FL, 1987.

61. Neuron Data Inc., "NEXPERT OBJECT Fundamentals", Palo Alto, CA, 1987.

62. Golden, M., Siemen, R. W. and Ferguson, J. C., "What's Wrong with Rules?", 1986 IEEE Western Conference on Knowledge-Based Engineering and Expert Systems, pp. 162 - 165, 1986.

63. Rauch-Hindon, W. B., "Problems and Paradigms", AI Expert, Vol. 3, No. 3, pp. 55 - 60, March 1988.

64. Goodwin, S. E. and Catani, M. J., "Simplified Thermal Design of Building Envelopes for use with ASHRAE Standard 90-75", Portland Cement Association, Skokie, Ill., 1976.

65. Analysis: Thermal Transfer through the Envelope, Architects Handbook of Energy Practice, The American Institute of Architects, 1982.

66. Fazio, P. and Gowri, K., "A Knowledge-Based System for the Selection and Design of Roof Systems", Building Research and Practice, Vol. 17, No. 5, pp. 294 - 298, 1989.

67. ASHRAE Handbook of Fundamentals, American Society of Heating Ventilating and Air-conditioning, New York, NY, 1981.

68. Yardsticks for Costing, Southam Building Reports, Toronto, Ontario, 1986.

69. Allen, E. and lano, J., "The Architect's Studio Companion: Technical Guidelines for Preliminary Design", John Wiley and Sons, New York, 1989.

70. Hardy, S., "Evaluating Roof Insulation Systems", Architecture, August 1985.

71. Nisson, J. D. N., "Windows and Energy Efficiency: Principles, Practice and available Products", Energy Design Update, Channers Publishing Company, MA, 1985.

72. Benoit, M., "Notes on Design Practice", private comunication, 1989.

73. Fazio, P. and Gowri, K., "Temperature and Vapor Pressure Gradients across Wall Assemblies using Microcomputer graphics", International Journal of Microcomputers in Civil Engineering, Vol.3, No.2, pp. 167 - 171, 1988.

74. Fazio, P. and Gowri, K., "BEADS - A Knowledge-Based System for Building Envelope Design", Proceedings of the First National Building Envelope Congress, Edmonton, Albetra, May 1989.

75. Fazio, P., Bedard, C. and Gowri, K., "Building Envelope Analysis and Design System", in Artificial Intelligence in Design, Proc. of the Fourth International Conference on the Applications of Artificial Intelligence in Engineering, Cambridge, UK, pp. 133 - 142, July 1989.

76. Fazio, P., Bedard, C. and Gowri, K., "Knowledge-Based System Approach to Building Envelope Design", Computer Aided Design, Vol. 21, No. 8, pp. 519 - 527, 1989.

77. Fazio, P., Bedard, C. and Gowri, K., "Constraints for Generating Building Envelope Design Alternatives", Evaluating and Predicting Design Performance, Proc. of the 3rd Int'l SUNY Buffalo CAD Symposium, Buffalo, NY, pp. 123 - 134, March 1990.

78. Allwood, R. J., Shaw, M. R., Smith, J. L., Stewart, D.J. and Trimble, E. G., "Building Dampness: Diagnosing the Causes", Building Research & Practice, Vol. 16, No. 1, 1988.

79. Fazio, P. and Gowri, K., "Knowledge-Based Design Assistant to Overcome Moisture Induced Problems", Thermal Performance of the Exterior Envelopes of Buildings IV, Proc. of the ASHRAE/DOE/BTECC/CIBSE Conference, Orlando, Florida, pp. 506 - 512, Dec. 1989.

80. Shaviv, E. and Kalay, Y. E., "Combined Procedural and Heuristic Method to Energy Conscious Building Design and Evaluation", Evaluating and Predicting Design Performance, Proc. of the 3rd Int'l SUNY Buffalo CAD Symposium, Buffalo, NY, pp. 245 - 260, March 1990.

APPENDIX - A

KNOWLEDGE BASE INFORMATION ON BASIC WALL AND ROOF TYPES

```
{{ BRICK-ON-CONCRETE-BLOCK-WALL
            IS-A:                      BASIC-WALL-TYPE
            NAME-OF-COMPONENTS:        BRICK
                                       AIR-SPACE
                                       INSULATION
                                       POLYETHYLENE
                                       CONCRETE-BLOCK
                                       GYPSUM-WALLBOARD
            R-VALUE:                   0.7925
            THICKNESS:                 337.5
            SUITABLE-STRUCTURE:        STEEL  COMPOSITE  CONCRETE
            SUITABLE-BUILDING-TYPE:    2  3  4  }}

{{ BRICK-ON-SHEATHED-STUD-WALL
            IS-A:                      BASIC-WALL-TYPE
            NAME-OF-COMPONENTS:        BRICK
                                       AIR-SPACE
                                       BUILDING-PAPER
                                       INSULATION
                                       POLYETHYLENE
                                       PLYWOOD-SHEATHING
                                       STRUCTURE
                                       GYPSUM-WALLBOARD
            R-VALUE:                   0.6044
            THICKNESS:                 152.65
            SUITABLE-STRUCTURE:        WOOD
            SUITABLE-BUILDING-TYPE:    1  2  3  }}
```

```
{{ BRICK-ON-STEEL-FRAME-WALL
                    IS-A:                     BASIC-WALL-TYPE
                    NAME-OF-COMPONENTS:       BRICK
                                              AIR-SPACE
                                              INSULATION
                                              POLYETHYLENE
                                              STRUCTURE
                                              GYPSUM-WALLBOARD
                    R-VALUE:                  0.4846
                    THICKNESS:                137.65
                    SUITABLE-STRUCTURE:       STEEL   COMPOSITE
                    SUITABLE-BUILDING-TYPE:   2   3   4  }}


{{ STONE-PANEL-ON-CONCRETE-BLOCK-WALL
                    IS-A:                     BASIC-WALL-TYPE
                    NAME-OF-COMPONENTS:       STONE-PANEL
                                              AIR-SPACE
                                              INSULATION
                                              POLYETHYLENE
                                              CONCRETE-BLOCK
                                              GYPSUM-WALLBOARD
                    R-VALUE:                  0.7432
                    THICKNESS:                287.5
                    SUITABLE-STRUCTURE:       STEEL   CONCRETE   COMPOSITE
                    SUITABLE-BUILDING-TYPE:   3   4  }}


{{ STONE-PANEL-ON-STEEL-FRAME-WALL
                    IS-A:                     BASIC-WALL-TYPE
                    NAME-OF-COMPONENTS:       STONE-PANEL
                                              AIR-SPACE
                                              INSULATION
                                              POLYETHYLENE
                                              STRUCTURE
                                              GYPSUM-WALLBOARD
```

```
                    R-VALUE:                0.4353
                    THICKNESS:              87.65
                    SUITABLE-STRUCTURE:     STEEL   COMPOSITE
                    SUITABLE-BUILDING-TYPE: 3   4   }}


{{ CONCRETE-BLOCK-WALL
                    IS-A:                   BASIC-WALL-TYPE
                    NAME-OF-COMPONENTS:     CONCRETE-BLOCK
                                            INSULATION
                                            POLYETHYLENE
                                            GYPSUM-WALLBOARD
                    R-VALUE:                0.5371
                    THICKNESS:              212.5
                    SUITABLE-STRUCTURE:     STEEL   CONCRETE   COMPOSITE
                    SUITABLE-BUILDING-TYPE: 2   3   4   }}


{{ METAL-SIDING-ON-CONCRETE-BLOCK-WALL
                    IS-A:                   BASIC-WALL-TYPE
                    NAME-OF-COMPONENTS:     METAL-SIDING
                                            AIR-SPACE
                                            INSULATION
                                            POLYETHYLENE
                                            CONCRETE-BLOCK
                                            GYPSUM-WALLBOARD
                    R-VALUE:                0.715
                    THICKNESS:              242.5
                    SUITABLE-STRUCTURE:     STEEL   CONCRETE   COMPOSITE
                    SUITABLE-BUILDING-TYPE: 2   3   4   }}


{{ METAL-SIDING-ON-STEEL-GIRT-WALL
                    IS-A:                   BASIC-WALL-TYPE
```

```
              NAME-OF-COMPONENTS:    METAL-SIDING
                                     AIR-SPACE
                                     INSULATION
                                     POLYETHYLENE
                                     STRUCTURE
                                     GYPSUM-WALLBOARD
              R-VALUE:               0.2292
              THICKNESS:             62.65
              SUITABLE-STRUCTURE:    STEEL   CONCRETE   COMPOSITE
              SUITABLE-BUILDING-TYPE: 2   3   4   }}


{{ PRECAST-CONCRETE-PANEL-WALL
              IS-A:                  BASIC-WALL-TYPE
              NAME-OF-COMPONENTS:    LIGHT-WT-CONCRETE
                                     INSULATION
                                     POLYETHYLENE
                                     STRUCTURE
                                     GYPSUM-WALLBOARD
              R-VALUE:               0.43
              THICKNESS:             162.65
              SUITABLE-STRUCTURE:    STEEL   CONCRETE   COMPOSITE
              SUITABLE-BUILDING-TYPE: 2   3   4   }}


{{ CONCRETE-WALL
              IS-A:                  BASIC-WALL-TYPE
              NAME-OF-COMPONENTS:    CONCRETE
                                     INSULATION
                                     POLYETHYLENE
                                     GYPSUM-WALLBOARD
              R-VALUE:               0.3416
              THICKNESS:             212.65
              SUITABLE-STRUCTURE:    CONCRETE   COMPOSITE
              SUITABLE-BUILDING-TYPE: 2   3   4   }}
```

```
{{ GLASS-FIBER-REINFORCED-CONCRETE-PANEL-WALL
                    IS-A:                      BASIC-WALL-TYPE
                    NAME-OF-COMPONENTS:        GLASS-FIBER-RC-PANEL
                                               AIR-SPACE
                                               INSULATION
                                               POLYETHYLENE
                                               STRUCTURE
                                               GYPSUM-WALLBOARD
                    R-VALUE:                   0.4071
                    THICKNESS:                 47.025
                    SUITABLE-STRUCTURE:        STEEL   CONCRETE   COMPOSITE
                    SUITABLE-BUILDING-TYPE:    4  }}

{{ STUCCO-ON-SHEATHED-STUD-WALL
                    IS-A:                      BASIC-WALL-TYPE
                    NAME-OF-COMPONENTS:        STUCCO
                                               GYPSUM-SHEATHING
                                               AIR-SPACE
                                               INSULATION
                                               POLYETHYLENE
                                               STRUCTURE
                                               GYPSUM-WALLBOARD
                    R-VALUE:                   0.5217
                    THICKNESS:                 68.15
                    SUITABLE-STRUCTURE:        STEEL   WOOD
                    SUITABLE-BUILDING-TYPE:    1   2   3  }}

{{ STUCCO-ON-CONCRETE-BLOCK-WALL
                    IS-A:                      BASIC-WALL-TYPE
                    NAME-OF-COMPONENTS:        STUCCO
                                               INSULATION
                                               POLYETHYLENE
                                               CONCRETE-BLOCK
                                               GYPSUM-WALLBOARD
```

```
        R-VALUE:                 0.5635
        THICKNESS:               230.5
        SUITABLE-STRUCTURE:      STEEL   CONCRETE   COMPOSITE
        SUITABLE-BUILDING-TYPE:  1   2   3  }}


{{ WOOD-SIDING-ON-SHEATHED-STUD-WALL
        IS-A:                    BASIC-WALL-TYPE
        NAME-OF-COMPONENTS:      PLYWOOD-SIDING
                                 BUILDING-PAPER
                                 FIBERBOARD-SHEATHING
                                 AIR-SPACE
                                 INSULATION
                                 POLYETHYLENE
                                 STRUCTURE
                                 GYPSUM-WALLBOARD
        R-VALUE:                 0.814
        THICKNESS:               70.65
        SUITABLE-STRUCTURE:      WOOD
        SUITABLE-BUILDING-TYPE:  1   2  }}


{{ WOOD-SIDING-ON-CONCRETE-BLOCK-WALL
        IS-A:                    BASIC-WALL-TYPE
        NAME-OF-COMPONENTS:      PLYWOOD-SIDING
                                 AIR-SPACE
                                 INSULATION
                                 POLYETHYLENE
                                 CONCRETE-BLOCK
                                 GYPSUM-WALLBOARD
        R-VALUE:                 0.8788
        THICKNESS:               255.65
        SUITABLE-STRUCTURE:      STEEL   CONCRETE   COMPOSITE
        SUITABLE-BUILDING-TYPE:  1   2  }}
```

```
{{ SPANDREL-PANEL-CURTAIN-WALL
        IS-A:                      BASIC-WALL-TYPE
        NAME-OF-COMPONENTS:        SPANDREL-GLASS
                                   AIR-SPACE
                                   INSULATION
                                   POLYETHYLENE
                                   STRUCTURE
                                   GYPSUM-WALLBOARD
        R-VALUE:                   0.4071
        THICKNESS:                 43.95
        SUITABLE-STRUCTURE:        STEEL   CONCRETE   COMPOSITE
        SUITABLE-BUILDING-TYPE:    4   }}


{{ SPANDREL-PANEL-ON-CONCRETE-BLOCK-WALL
        IS-A:                      BASIC-WALL-TYPE
        NAME-OF-COMPONENTS:        SPANDREL-GLASS
                                   AIR-SPACE
                                   INSULATION
                                   POLYETHYLENE
                                   CONCRETE-BLOCK
                                   GYPSUM-WALLBOARD
        R-VALUE:                   0.715
        THICKNESS:                 243.75
        SUITABLE-STRUCTURE:        CONCRETE   COMPOSITE
        SUITABLE-BUILDING-TYPE:    4   }}


{{ SHINGLE-OR-SLATE-ON-WOOD-DECK
        IS-A:                      BASIC-ROOF-TYPE
        NAME-OF-COMPONENTS:        ASPHALT-SHINGLES
                                   INSULATION
                                   POLYETHYLENE
                                   WOOD-DECK
                                   WOOD-RAFTER-2X8@400MM
                                   GYPSUM-WALLBOARD
```

```
                         R-VALUE:                 1.1021
                         THICKNESS:               284.15
                         SUITABLE-STRUCTURE:      WOOD
                         SUITABLE-ROOF-SHAPE:     PITCH>3/12  }}


{{ SHINGLE-OR-SLATE-ON-SHEATHED-RAFTER
                         IS-A:                    BASIC-ROOF-TYPE
                         NAME-OF-COMPONENTS:      ASPHALT-SHINGLES
                                                  INSULATION
                                                  BUILDING-PAPER
                                                  PLYWOOD-SHEATHING
                                                  WOOD-RAFTER-2X8@400MM
                                                  GYPSUM-WALLBOARD
                         R-VALUE:                 0.5615
                         THICKNESS:               224
                         SUITABLE-STRUCTURE:      WOOD
                         SUITABLE-ROOF-SHAPE:     PITCH>3/12  }}


{{ BUILT-UP-OR-SINGLE-PLY-ON-METAL-DECK
                         IS-A:                    BASIC-ROOF-TYPE
                         NAME-OF-COMPONENTS:      STONE
                                                  THREE-PLY-BUILT-UP-ROOF
                                                  INSULATION
                                                  GLASS-FIBER
                                                  POLYETHYLENE
                                                  METAL-DECK
                                                  GYPSUM-WALLBOARD
                         R-VALUE:                 0.9119
                         THICKNESS:               116.65
                         SUITABLE-STRUCTURE:      STEEL   CONCRETE
                         SUITABLE-ROOF-SHAPE:     FLAT   PITCH<3/12  }}
```

{{ BUILT-UP-OR-SINGLE-PLY-ON-POURED-CONCRETE-SLAB
IS-A: BASIC-ROOF-TYPE
NAME-OF-COMPONENTS: STONE
THREE-PLY-BUILT-UP-ROOF
INSULATION
POLYETHYLENE
CONCRETE-SLAB
GYPSUM-WALLBOARD
R-VALUE: 0.3287
THICKNESS: 122.65
SUITABLE-STRUCTURE: COMPOSITE CONCRETE
SUITABLE-ROOF-SHAPE: FLAT PITCH<3/12 }}

{{ BUILT-UP-OR-SINGLE-PLY-ON-PRECAST-CONCRETE-SLAB
IS-A: BASIC-ROOF-TYPE
NAME-OF-COMPONENTS: STONE
THREE-PLY-BUILT-UP-ROOF
INSULATION
POLYETHYLENE
HOLLOW-CORE-SLAB
GYPSUM-WALLBOARD
R-VALUE: 0.7478
THICKNESS: 222.65
SUITABLE-STRUCTURE: STEEL CONCRETE COMPOSITE
SUITABLE-ROOF-SHAPE: FLAT PITCH<3/12 }}

{{ BUILT-UP-OR-SINGLE-PLY-ON-WOOD-DECK
IS-A: BASIC-ROOF-TYPE
NAME-OF-COMPONENTS: STONE
THREE-PLY-BUILT-UP-ROOF
INSULATION
POLYETHYLENE
WOOD-DECK
WOOD-JOIST-2X10@400MM

```
                               GYPSUM-WALLBOARD
       R-VALUE:                1.0827
       THICKNESS:              332.65
       SUITABLE-STRUCTURE:     STEEL   WOOD
       SUITABLE-ROOF-SHAPE:    FLAT   PITCH<3/12  }}


{{ BUILT-UP-OR-SINGLE-PLY-ON-GYPSUM-DECK
       IS-A:                   BASIC-ROOF-TYPE
       NAME-OF-COMPONENTS:     STONE
                               THREE-PLY-BUILT-UP-ROOF
                               INSULATION
                               POLYETHYLENE
                               GYPSUM-DECK
                               WOOD-JOIST-2X10@400MM
                               FORMBOARD
       R-VALUE:                1.2887
       THICKNESS:              320.15
       SUITABLE-STRUCTURE:     STEEL   WOOD   COMPOSITE
       SUITABLE-ROOF-SHAPE:    FLAT   PITCH<3/12  }}


{{ METAL-PANEL-ON-SHEATHED-RAFTER
       IS-A:                   BASIC-ROOF-TYPE
       NAME-OF-COMPONENTS:     SHEET-METAL-ROOFING
                               INSULATION
                               BUILDING-PAPER
                               PLYWOOD-SHEATHING
                               WOOD-RAFTER-2X8@400MM
                               GYPSUM-WALLBOARD
       R-VALUE:                0.634
       THICKNESS:              287.5
       SUITABLE-STRUCTURE:     STEEL   WOOD
       SUITABLE-ROOF-SHAPE:    FLAT   PITCH<3/12  }}
```

```
{{ METAL-PANEL-ON-WOOD-DECK
                        IS-A:                   BASIC-ROOF-TYPE
                        NAME-OF-COMPONENTS:     SHEET-METAL-ROOFING
                                                INSULATION
                                                BUILDING-PAPER
                                                WOOD-DECK
                                                WOOD-JOIST-2X10@400MM
                                                GYPSUM-WALLBOARD
                        R-VALUE:                1.1852
                        THICKNESS:              350.0
                        SUITABLE-STRUCTURE:     STEEL   WOOD
                        SUITABLE-ROOF-SHAPE:    FLAT   PITCH<3/12  }}
```

```
{{ SHEET-METAL-ON-PURLIN
                        IS-A:                   BASIC-ROOF-TYPE
                        NAME-OF-COMPONENTS:     METAL-ROOF-PANEL
                                                INSULATION
                                                METAL-DECK
                                                GYPSUM-WALLBOARD
                        R-VALUE:                0.5142
                        THICKNESS:              113.5
                        SUITABLE-STRUCTURE:     STEEL   WOOD
                        SUITABLE-ROOF-SHAPE:    FLAT   PITCH<3/12  }}
```

APPENDIX - B

DQL KEYWORDS LIST

ACTION KEYWORDS:	CONSULT
	GENERATE
	EVALUATE
	ADD
	DELETE
	DISPLAY
	MODIFY
	FIND
QUALIFIERS:	BASIC
	FEASIBLE
	GROSS
	PERMISSIBLE
OBJECTS:	WALL
	ROOF
	GLAZING
	INSULATION
	BUILDING
	FENESTRATION
	ALTERNATIVE
ATTRIBUTES:	AREA
	THICKNESS
	LOCATION
	TYPE
	R-VALUE
	COST
OPERATORS:	<=
	=
	>=