A CONTRIBUTION TO
ARCHITECTURAL / ENGINEERED DESIGN FOR TIMBER STRUCTURES
USING KNOWLEDGE-BASED METHODS

by

ROBERT JOHN TAYLOR

Dipl. Civil Engineering Technology, Ryerson Polytechnical Institute, Toronto, 1976
B.Sc.(Eng.)(Hons.), Queen's University at Kingston, 1982
M.Sc.(Eng.), Queen's University at Kingston, 1983

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

Department of Civil Engineering

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

October 1995

Department of Civil Engineering

The University of British Columbia
2324 Main Mall
Vancouver, B.C.
Canada V6T 1Z4

11 October 1995

Abstract

This thesis attempts to synthesize knowledge from the fields of architecture, engineering, and computer science in the context of design. In particular, a novel approach to modeling the architectural and engineering design of structural connections is presented. Computer automation using parametric object-oriented methods for quantitative design is new for connections, and the inclusion of qualitative features native to architectural considerations presents a more holistic view to automated design of connections.

A unique method of representing connections as a kit of parts for assembly is presented that is based on the load path within the connection. The configuration model facilitates engineering discretization and evaluation; while the connection, if properly designed, can be more easily "read" by the observer - a desirable feature of a good work of architecture.

Quantitative aspects, typically thought of as engineering qualities, are combined with the adapted qualitative, typically architectural, aspects of a designed artifact through the use of dynamic fuzzy logic membership functions. A fuzzy logic adaptation of the qualitative attributes of a designed artifact can be used for assessing or generating aesthetics consistent within the scope of aesthetic definitions offered by the designer. The adaptation, therefore, does not constrain the designer to a prescribed attribute definition, but an architectural expression which is personal and unique. A brief development of membership function representation, calibration, and application is offered. Results from a particular demonstrative study of proximity, and another on colour reveal a promising application of fuzzy logic technology to qualitative design issues.

Among a number of smaller innovations, the main contribution of this thesis to the advancement of knowledge is three fold: a new method to represent structural connections in general; a synthesis of truths underlying connection configuration design in timber structures so that design automation using object-oriented methods can be facilitated; and development of an automation method for connection design that separates program control from object data, which is a significant benefit in ease of automated application expansion. The

work presented here is intended to break new ground in these areas for others to investigate further towards resolving a significant need in design.

Contents

Tables

Figures

Notation

Chapter 3

A, B, C	=	descriptive variable name;
$C(x)$	=	degree of certainty at a value x;
$C(x)_{Mi}$	=	degree of certainty at a discrete value of x of the ith membership function M;
M_i	=	the ith membership function;
n	=	numerical quantity;
$o(v)$	=	raw classification occurrences data at discrete values v;
$P1,P3$	=	value point at 5th percentile of distribution of $0<T<1$;
$P2,P4$	=	value point at 95th percentile of distribution of $0<T<1$;
$S1,S2$	=	limb slope of trapezoid-shaped membership function;
T	=	degree of truth, degree of membership;
$T(x)$	=	degree of truth (membership) at a value x;
$T(x)_{Mi}$	=	degree of truth at a discrete value of x of the ith membership function M;
v	=	measurable quantity, value;
X	=	name of a design space, domain;
x	=	input value;
$\overline{x}$	=	mean of x values;
y	=	input value; and
σ	=	standard deviation.

Chapter 5

A	=	applied axial force at member end;
A_β	=	applied axial force at member end aligned parallel with wood grain;
b	=	length parallel to the member axis of a grid of connectors;
C	=	connector group centroid location;
d	=	bolt shank diameter (mm);
e	=	eccentricity taken as the distance from the centroid of the connector group to the end of the member;
f	=	wood embedment strength of main member by bolt shank (MPa);
F_1	=	reduced bearing strength of steel side plate (MPa);
f_1	=	bearing strength of steel side plate as noted (MPa);
f_1	=	embedding strength of wood side member as noted (MPa);
f_2	=	embedding strength of main member (MPa);
f_y	=	bolt yield strength (MPa);
h	=	length perpendicular to the member axis of a grid of connectors;
i	=	subscript;
J	=	polar moment of inertia of the strengths S about the centre of rotation (the fastener group centroid);
J_C	=	row factor for up to 12 bolts in a row;
J_F	=	joint capacity reduction factor;
J_L	=	factor for number of rows;
K_D	=	wood material modification factor for load duration;
K_{SF}	=	wood material modification factor for service condition;
K_T	=	wood material modification factors for fire treatment;
l	=	length of a row of connectors;
l	=	member thickness (mm);
l_1	=	steel side plate thickness as noted (mm);
l_1	=	wood side member thickness as noted (mm);
l_2	=	main member thickness (mm);

m	=	number of member-axis-perpendicular rows in a grid of connectors;
M	=	applied moment at member end;
M_C	=	total applied moment located at connector group centre of rotation;
n	=	number of member-axis-parallel rows in a grid of connectors;
N	=	number of fasteners in the group;
n	=	number of fasteners in the fastener group;
N	=	number of bolts in row;
n_f	=	number of fasteners in a group;
n_s	=	number of shear planes across the bolt;
P_r	=	lateral shear strength in wood parallel to the grain of a group of bolts;
P_u	=	factored lateral shear strength in wood parallel to the grain of a single bolt (kN);
p_u	=	specified lateral shear strength in wood parallel to the grain of a single bolt (N);
Q_r	=	lateral shear strength in wood perpendicular to the grain of a group of bolts;
Q_u	=	factored lateral shear strength in wood perpendicular to the grain of a single bolt (kN);
q_u	=	specified lateral shear strength in wood perpendicular to the grain of a single bolt (N);
r	=	radius from connector group centroid to connector centroid;
R	=	resultant fastener resisting force;
S	=	allowable connector shear load;
S	=	scale up/down factor of 3D object;
s	=	in-row spacing (mm);
T	=	fastener moment resistance force in the wood;
T_x	=	translation factor of 3D object in x direction;
T_y	=	translation factor of 3D object in y direction;
T_z	=	translation factor of 3D object in z direction;
V	=	applied shear force at member end;
V_β	=	applied shear force at member end aligned perpendicular to wood grain;
X,Y	=	2D point coordinates in picture plane;
x, y	=	corresponding connector centroid x and y coordinates, respectively;
x,y,z	=	3D vertex point coordinates of 3D object;
α	=	radial angle to the grain of each resultant fastener resisting force (degrees);
β	=	angle that wood grain angle makes to member axis (degrees);
γ	=	angle from horizon to member axis (degrees);
θ	=	estimated angle from wood grain to fastener force (degrees);
θ_1	=	direction of load angle to side member grain (degrees);
θ_2	=	direction of load angle to main member grain (degrees);
θ_h	=	horizontal viewing angle to 3D object z-axis in x-z plane;
θ_v	=	vertical viewing angle to 3D object x-z plane;
ρ_k	=	characteristic wood density (kg/m^3); and,
ϕ	=	performance factor.

Acknowledgements

The awareness of the need for automated connection design research became apparent to me in professional practice many years ago. This thesis marks a beginning in resolving a significant and serious market need in the design world, an interesting academic challenge, and I hope others will carry it on. There are many people to thank: those in practice that constantly reminded me of the need, and those who supported me in the cause of dealing with it.

Financial support is gratefully acknowledged through fellowships and research associateship with the Civil Engineering Department of UBC, Forintek Canada Corporation, Forestry Canada, and NSERC. I am also grateful for the academic support and encouragement from the Civil Engineering Department, the School of Architecture, and the UBC Faculty of Graduate Studies in the completion of this unique PhD program. The many students who have crossed my path in my teaching assistant duties in Architecture and Civil Engineering have provided me with interesting insights and aspects to design problems.

By name in particular, I would like to thank my supervisor Dr. Siegfried Stiemer (UBC Civil) and Dr. Erol Varoglu (Forintek) for their invaluable research support, criticism, and friendship. Dr. Ricardo Foschi (UBC Civil) kindly provided invaluable program advice and encouragement early in this endeavour as well as computer equipment for the program's duration. Similar advice, encouragement, and guidance was given by Mr. Stephen Taylor (UBC Architecture) and Dr. Helmut Prion (UBC Civil) who always provided interesting and considerate points of view. My committee members are to be thanked for their reading and constructive criticism of this work. The efforts of Mrs. Anne Miele, and the UBC Library staff in providing sourcing assistance for many difficult-to-find literary publications cited in this thesis are also acknowledged.

Recognition to Mr. and Mrs. Floyd Cormin, and Mr. and Mrs. John W. Taylor is to be given for their kindly grateful and generous contribution in patience, encouragement, and financial support. Finally, loving gratitude is extended to my wife Shurli, and sons Isaac and Aren, for whose limitless love, patience, tolerance, and support through this whole experience was the difference that made this academic goal possible for me.

Robert J. Taylor

Chapter 1 INTRODUCTION

1.1 BACKGROUND

Forests are renewable resources that can be productively, responsibly, and economically farmed continuously over time to manufacture various wood products. As a result of increasing environmental concerns of public and regulatory agencies, wood products have the potential of becoming increasingly competitive with other building materials. The recent development of engineered wood products of low variability in strength and stiffness, introduction of new types of connections and fasteners, and new techniques in timber construction as outlined in Goetz *et al.* (1989), are major factors that turn engineers and architects to timber construction in several situations as a viable alternative to construction in masonry, reinforced concrete, or steel.

Connections in architecture and structural design are of focal importance in the context of the total design process and are often the source of grief for many a designer, especially when detailing. Although connections between members are necessary for many modular materials, timber connections probably are the most interesting (because of the combination of materials and available fastenings employed) and the most difficult to detail properly. In many timber frame buildings, each of the structural components requires connections with others designed to suit the expected function of the structure and often the aesthetic theme of the building, as well as to provide structural safety. In many cases, the performance of the building is not only governed by the characteristics of structural components, but also the way these components are connected together. Market studies have identified that one of the major impediments to expanded use of wood products in low-rise non-residential construction is the complexity and cost of current practice in timber connection design. To the architect, wood is a material that wants to be expressed because of its natural beauty and patina, the expressive quality of which can only be enhanced by the appropriate attention to joint and connection detail. If a number of one-off connection designs are employed in a timber project, the design cost can quickly escalate to the point of jeopardizing the project's economic viability. A way to deal with the complexity of holistic timber connection design is by developing a design aid for use on a personal computer by an architect or engineer, which is the focus of this thesis.

In addition to timber joinery, connecting timber members is traditionally accomplished in basically three ways: by fasteners alone, by fasteners and off-the-shelf hardware, and by fasteners and custom-fabricated hardware. While the first two methods use engineering design information primarily from timber codes or hardware manufacturer literature, the latter method requires additional information from engineering design codes in steel. Hence, a knowledge-based system for timber connection design presents an excellent opportunity to formalize knowledge in this area as well as facilitate architectural and engineering design with a traditional renewable building material resource. The applied research offered by this thesis is novel in its approach to combining design abilities of the professions of architecture and structural engineering, enabling communication between the architect and engineer, and in the development of computer technology for the yet un-automated design of timber connections.

1.1.1 Architectural and Engineering Design

A designer works at the edge of the known and the possibilities of that which may be. Building designers commonly fall into two categories: architecture or engineering. Most dictionaries define an engineer as a person trained to employ a science by which the properties of matter and sources of energy are made useful to man in structures, machines, and products. In the context of building design, the engineer is professionally responsible for structural integrity of the building in an economical way. Similarly, dictionaries define the architect as a person who is professionally concerned with the forms of buildings, with the functions of buildings, and with how to relate the two. This definition includes semantic and symbolic aspects of form, interior space and exterior mass disposition to satisfy functional criteria, suitability of form to context, and social implications of form.

Although architectural and engineering designers both deal with building, they each think very differently about the same subject matter. One of the substantive differences between the two professions noted by Benjamin (1984) and Lin and Stotesbury (1988) is that engineers generally use the abstract language and notation of mathematics to develop their designs, whereas architects generally employ visual and graphic notation. Engineering and architectural modes of thinking are both complex. One mode, the scientific, is hierarchically ordered, or organized in vertically logical systems typified by strategic thinking or planning of processes. The other mode, the empirical, is associative or laterally organized which appears in the deceptively random-seeming pragmatism of design or manufacture. Design, utilizes lateral thinking. Lateral thinking is almost exclusively synthetical in purpose rather than analytical. It synthesizes or creates as a primary activity rather than dissects. Analysis, which is a vertical thought process, is useful as a check in the development of design and for reflection. In contrast to scientific thought, which follows well-defined methods, there is no formal balance between synthetical and analytical thinking, but only an unstable mixture which can shift. It is open-ended and subject to interpretation. Engineering and architectural designers experience this tense polar malaise between the so-called cultural and the technological world. It feels like a basic flaw in these professions, a hiatus in thinking, and they some guiltily feel that they ought to be able to reconcile the two. Historically, design thinking was not always so diverse, and it may be moving closer again through the use of the computer as noted by the Building Arts Forum (1991). Each mode of thinking can enrich the other, and architects and engineers could profit from an exchange of viewpoint and perhaps even understand each other, if an effort to permit this to happen is advanced.

An architectural design must incorporate a structural system or systems that will help the architect transform the design conveniently, safely, and economically, into a structure that resists environmental elements and encloses spaces according to his original ideas while meeting the needs of a client. Architects need a knowledge of structures to ensure that the practical expressions of their architecture are both rational and efficient. An improper choice of structure may work against an architect. For example, an architect who wishes to create an impression of airiness may discover later that the architecture actually expresses monolithic massiveness - all by an improper choice of structure or materials.

There is a strong belief that since architects normally work with consulting structural engineers who make the final decisions on the structural system and the sizes to be used, a knowledge of structures for architects is not really necessary. A contrary belief for engineers vis-á-vis architecture is also held. Two practical arguments illustrate the need for further study of this interface between structure and architecture. First, an architect shy of structural principles often hands over a beautiful piece of architecture to a possibly unthinking, unfeeling engineer who then proceeds to insert a structural system into it. The engineer's decisions flow from his/her own philosophy of training, including such factors as pure economics and ease of calculation and construction, and they may express something quite different from the architect's overall philosophy. Since the structural system and materials of construction largely dictate the building's appearance, the architect has permitted the engineer to "suppress" the architecture. The second argument is that an architect lacking a structural understanding may propose a piece of architecture that cannot be built at a price the client is prepared to pay; something beyond being reasonably practical or even feasible.

The architect who understands structures and uses this knowledge from the initial stages of design will produce a feasible and practical design. Furthermore, in his consultations with the structural engineer, he will be able to discuss architectural objectives intelligently and to appreciate, or if necessary resist, suggested changes. This architect operates from a better position of knowledge and strength, which with the structural engineer, will produce a better design faster and with less cost.

Structure should influence architecture by making it more rational and efficient, to the point that it may dictate architecture. Architects need to realize that, although the architectural design has spans or spaces of its own, the spans they impose on the structural system influence the choice of that very system. Equally important is the realization that the architectural design, by its shape in plan and elevation, and by the objective it tries to achieve, can suggest other systems better suited, or worse, to the design itself. The integration of any structure into the architectural design is always desirable. For very large structures, bridges, or space structures, usually integration is best done by designing the architecture around the structure itself. In these structures, the domination of the visual architectural expression by structure is almost total and complete, a domination which can only be either subdued or emphasized but not eliminated. Whether architecture loses or gains by the domination of structure is a matter that the individual architect, motivated by his own design philosophy, must decide for himself.

Architects then mainly concern themselves with the visual impact and meaning of structure, and generally employ graphic notation and visual means to do their work. Architectural design thinking is typically associative or lateral in nature. Engineers concern themselves more with practical issues of safety and economy, generally using the abstract language and notation of mathematics to develop their designs. Engineering design thinking is typically linear, hierarchically ordered, or organized in vertically logical systems typified by strategic thinking or planning of processes. Such diverse thinking processes present an interesting challenge for integration in a more holistic design process.

The abstract language engineers work in is mainly numerical, while the associative language of architecture is expressed more commonly graphically, but also linguistically in which the linguistic can be either explicit (quantitative) or vague (qualitative). Such numerical, graphical, and linguistic information forms the basis of description in the design world.

1.1.2 The Design Process

The design process has frequently been modelled as recursive interaction of the activities of analysis, synthesis, and evaluation as shown in Figure 1.1. Tang (1982) offers an architectural insight of considerable depth into the design process which generally corresponds closely to that of Figure 1.1. During the development of a design;

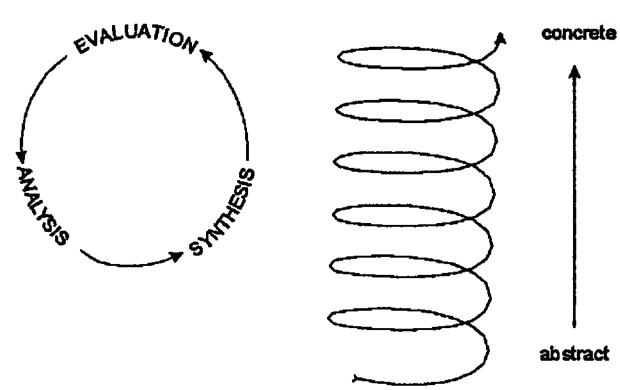


Figure 1.1 The Design Process Model

Woodbury (1987), Radford and Gero (1988), and Mitchell (1990) agree that the progress towards a solution is achieved by generating new states from current ones. While not a *well-conditioned problem* type, such problems which information is accumulated during the process of solution are representative of a class of problems referred to as *ill-defined problems* (Oxman and Gero 1987), or worse, *wicked problems* (Bazjanac 1974). These three problem types are more informally described by these authors as:

- *Well Conditioned* - a direct solution is obtained from a set of well-described facts, processes, and unknown variables.
- *Ill-Defined* - solution to the problem is a function of the *statement* of the problem, i.e. is the problem to know what is wrong, or is it to rectify it; and in this context, what does "rectify" really mean? To solve, it is necessary to know:
    1. how to decompose the ill-defined problem into well-defined parts,
    2. how to resolve the well-defined parts, and
    3. how to reassemble these partial solutions into a general solution for the entire problem (problem structuring / puzzle-making).
- *Wicked Problems* - a class of problems which are ill-formulated, where the information is confusing, where there are many participants and decision makers with conflicting values, and where the ramifications in the whole system are thoroughly confusing. Table 1.1 summarizes eleven properties of wicked problems.

Design problems turn out to be classed as wicked problems. Design then can be interpreted in a state-space representation as an initial state which is transformed using expert knowledge into a series of solution states (Oxman and Gero 1987). Problem-solving also can be seen as a process of searching through alternative solution states which satisfy certain goals, and thus shares many of the characteristics of the design problem. Archea (1987) believes that architects engage in more than just problem-solving:

> *puzzle-making*: Instead of specifying what they are trying to accomplish prior to their attempts to accomplish it as problem-solvers do, architects treat design as a search for the most appropriate effects that can be attained in a unique context. They seek sets of combinatorial rules that will result in an internally consistent fit between a kit of parts and the effects that are achieved when those parts are assembled in a certain way.

Table 1.1 Eleven Properties of Wicked Problems

Property (1)	Description (2)
1.	Wicked problems have no definitive formulation.
2.	Every formulation of the wicked problem corresponds immediately in mind to the formulation of the solution (and vice versa).
3.	Wicked problems have no stopping rule. Anytime a solution is reached, it could be improved or worked on some more.
4.	Solutions to wicked problems cannot be true or false, only "bad" or "good".
5.	In solving wicked problems, any conceivable plan, strategy or act is permissible in finding a solution and none can be prescribed as mandatory.
6.	For every wicked problem, there is always more than one possible explanation.
7.	Every wicked problem is a symptom of another, "higher level" problem.
8.	No wicked problem and no solution to it has a definitive test.
9.	Each wicked problem is a one-shot operation.
10.	Every wicked problem is unique. No two problems are exactly alike and no solutions or strategies leading to solutions can readily be copied for the next problem.
11.	The wicked problem solver has no right to be wrong -- he/she is fully responsible for their action.

One of the characteristics of architectural design is that of problem solving in a semantically rich task domain. Semantics, here, is defined as the creation of suggestions of meaning along with, or in addition to, the exact explicit meaning of a designed artifact. When problem solving in semantically rich domains, a large part of the problem solving search involves the use of long term human memory and is guided by information discovered in that memory; information that relates meaning to physical features of the artifact. The semantic representation of design knowledge provides a potential basis for both the organization of design knowledge, as well as for the definition of goals, or the provision of control in search procedures. Designers express their thoughts, concepts, and designs both verbally and graphically. An intelligent system for design in a computerized environment should provide the means to map between both sets of communications. In order to accomplish this, a semantic model must be constructed which is capable of recognizing the syntactic attributes of the geometrical description of objects and their relationships. Conversely, the syntactic geometric model must be capable of following semantic descriptions and operations. Both methods of manipulating ideas and knowledge are characteristic of the way designers work during the process of design. Sketching in design for example, and particularly in architecture, offers an exploratory medium through which the semantic attributes of the design emerge which were not previously anticipated.

According to Oxman and Gero (1987), knowledge in design may be briefly introduced as the tool whereby

- the designer conceptualizes the semantic content of a certain domain, and by which,
- the designer represents his ideas about that domain using syntactical relations between *objects* (domain knowledge), and *actions* manipulating these objects and relationships (control knowledge).

To clarify; semantic concerns deal with meanings of objects; for example, relationships between objects and between their attributes. Semantic facts about relationships between objects may include: positional relations (window is *above* sink), inclusion relations (window is *in* north wall), comparative relations (west wall is *short*), relations which define a ratio of states (area of kitchen has a proportion between length and width), and so on. Syntactic aspects deal only with physical descriptions of objects and geometrical connections to their domain or other objects; data which supports facts. Meanings underlying or evoked by object descriptions and geometrical connections are not involved. For example, when a designer graphically locates a window object in a wall object, syntactic knowledge is used to find the horizontal and vertical distances from the wall boundaries - discrete factual syntactic quantities. Knowledge in design, then, can be described symbolically (semantically) as well as mathematically (syntactically). A knowledge-based view of design intends to render knowledge through which design solutions are generated explicit and amenable to the process of computation.

1.1.3 Anatomy of Structures

Aristotle (1957) noted that physical objects in the world have form, function, and behaviour. Formal properties of an object can be classified in terms of geometry (shape and position in space), and material (material physical and engineering properties, surface appearance, and installed cost). The functions of many structural objects are to transfer load, and provide visual expression. The behaviour of these objects represents the internal response mechanism to external stimuli, such as applied load (magnitude, direction, location and direction, type).

1.1.4 Importance of Connections

Connections between timber, steel, or concrete structural members are important to design, and lack of proper attention to the detailing of connections can easily convert a good design into a bad one. It is important to detail the connection so that assumptions and intentions made in the design are realized. Space frame structures consist of a large number of members connected at joints, which may be quite significant in number. An inefficient connection, either in an aesthetic or in an engineering or economic sense, being multiplied so many times over can completely destroy the efficiency and architecture of the structure. Moreover, connections, particularly in timber, can be tedious and time consuming to design which may have significant associated design costs if the space frame is of random geometry consisting of many different joint conditions. Benjamin (1984) believes that good connections are vital to good design; the key components being aesthetics, structural capacity, fire resistance, and economy of means. The variety of components directly requires design teamwork between the architect and structural engineer at the detailed design stage, so that the overall design intentions for the project are realized in a holistic sense.

The aim of holistic connection design in general is to join objects together safely, economically, and attractively. In terms of timber connection design, safety issues such as: compatibility (fit of parts), capacity, fire, and stability (wood shrinkage and possible member movement) must be addressed. Economy issues usually focus on a combination of connection cost and performance, while attractiveness looks at the connection aesthetic and expression. The amount and diversity of information dealt with in good connection design presents a time consuming and costly exercise for any designer, thus creating considerable designer fear when faced with the task.

1.1.5 Automated Design Methods for Connections

Design can be considered a creative process, and the most difficult form of creativity is common sense. The more obvious things are to people, the harder it seems to be able to explain them to computers. Lawson (1990) claims that, because of its basic nature, a computer can only do three things: compute functions, solve problems, perform simulations, *and nothing else*. Thus, any task to be computer implemented must be formulated in terms of any of these three conditions. Taylor (1988) believes that computers are completely noncreative, so lack of creativity in solving problems is an essential approach to applied computer solutions . He claims that expert systems have no ability to develop new solutions to problems, and thus do not exhibit intelligence at all. His opinion is based on the view that reasoning is only based on the mechanical application of rules of logic to facts, and any problem that cannot be solved "by the book" is out of bounds. The more possibilities an expert has to consider, the better candidate a domain is for expert systems. Computers can be better than people at searching through every fact to make sure no possibility is overlooked.

Expert systems can capture valuable human expertise that is scarce or expensive or both. Rychener (1985) defines their characteristics as having:

- the ability to explain and justify answers on the basis of theory, heuristics, or case history,
- the closeness of reasoning procedures to those used by human experts,
- the ability to deal with uncertain or incomplete information,
- the ability to summarize and point out features of the problem situation that were most important in leading to an answer,
- the use of verbal or symbolic encoding for knowledge communicated in a natural way, and
- the ability to grow gradually by adding new pieces of knowledge usually in the context of solving an unfamiliar problem.

Ideally, if expert knowledge is captured in computer software, fewer people will have to learn to solve a given sophisticated problem in the future, and thus the software has some measure of industrial significance. However, the main reason (usually with explicit intention) for the development of expert systems according to Taylor (1988) is to save money and time, and often the advantages of an expert system are significant enough to justify a major effort to build them. The current trend of people becoming costlier and computers becoming cheaper may continue to result in many future opportunities for expert systems in general.

An expert system as defined by Taylor (1988) is collected rules of thumb; a knowledge system consisting of facts (body of information) and heuristics (methodological statements and rules of good guessing) which together characterize expert level decision making within a field - *human expertise in a computer*. Rules are useful in the making of fast decisions towards satisfaction of a goal, while heuristics are used to prune the search for the appropriate route towards satisfying the goal. The operating cycle of an expert system is straightforward:

get the facts, see if any rules fire, take the action implied by whichever rule fires, acquire more facts, and repeat until done. Taylor (1988) cites five major parts of an expert system:

(1) *user interface* - obtains information from the user, or sensor equipment;

(2) *database* - factual information stored about a specific problem. Facts are a list of terms that are related in some way and can be stored in simple lists or relational databases. Speed of data retrieval is an important consideration if the expert system is to run at an acceptable speed;

(3) *rule control structure* - decides the order in which rules are examined and the order in which questions are asked in order to obtain facts. Backward-chaining systems, which work backward from the goal such as diagnostic systems, tend to produce well-focused dialogues with users because all the questions about facts needed to satisfy a rule are asked at the same time. Forward chaining systems, which work from available facts toward an unknown goal and test the rules to see if their conditions are met, seem to jump around because the system does not readily settle on a direction. Engineers facing a problem, often gather facts about the problem, forward chain from the facts to guess what the problem might be, and then backward chain to test the hypothesis;

(4) *rule system* or *rule interpreter* or *inference engine* - rules are processed by the inference engine when the control structure decides that a fact would be of interest to the consultation with the user. Rules are written and debugged by the expert system developer, while the other parts of the system: inference engine, control structure, database, and interface are usually purchased as parts of an expert system tool kit called an *expert system shell*; and

(5) *rules* - knowledge engineering is the process of converting human knowledge into expert system rules. Each rule, which contains general knowledge about the domain, has to be understood, coded, entered into the computer, compiled, tested, debugged, documented, and subjected to version control. Rules operate differently from functions. Instead of being examined when control is passed by a calling function, rules "operate" when information becomes available. Otherwise, debugging rules of an expert system for unexpected interactions is like debugging conventional software.

Three fundamentally different kinds of expert systems cited by Taylor (1988) are:

- *rule-based* - rules encode experiential observations and are purely empirical in that the expert system knows nothing of any underlying causality. Such systems are called *shallow systems* and are said to use *shallow reasoning*;
- *model-based* - supplement empirical rules with knowledge about the real world. Many trouble shooting systems are of this type which, having guessed a fault from initial symptoms, the expert system simulates the fault in a real world model in order to discover more symptoms to prove or disprove the guess; and
- *knowledge-based* - one of the major difficulties in operating expert systems is choosing the next rule to examine. Knowledge-based systems have rules to tell which rules to look at next. Model-based and knowledge-based expert systems are sometimes called *deep reasoning systems*.

Oxman and Gero (1987) believe traditional design expert systems come in two types: *synthesis*, where the system capable of design generation; and *diagnosis*, where the system functions as a design critic to evaluate, criticize and recommend corrections in design. The key feature is that the knowledge base and control mechanism are separated in such a way as to recognize the *recursive nature* and *multiple modes* of design paradigms. According to Oxman and Gero (1987), there are three ways in which a user may interact with a design expert system: as a design diagnosis system used to check a designer-completed design described graphically; a design development system where the designer develops a partial design, has it checked by the system, and then interacts with it to generate a completed design which satisfies the implicit constraints from this partial initial state; or, a design generation system where the designer interacts with system from the beginning of the design process and generates a completed solution utilizing the knowledge in the system to satisfy the goals and constraints.

Further, Oxman and Gero (1987) give three types of knowledge significant in a design expert system:

1. *Syntactical knowledge* - often geometrical in nature, is concerned with and serves to maintain the connection of an object with its domain or other objects and with the data which supports facts;
2. *Semantic knowledge* - concerned with the meaning of objects and normally deals with the relationships between objects and between their attributes; and

3. *Performance or evaluation knowledge* - knowledge to interpret a design in terms of implicit attributes, often by knowledge-driven procedural programs. It is used to check the validity of a state solution against the system's knowledge base of performance requirements.

The utilization of these three types of knowledge and their significance in the operation of such an expert system for design is demonstrated by Oxman and Gero (1987) by a typical predicate of the system:

```
CHECK-EXECUTE:
        locate object
        generate facts
        interpret (find 'the approval of the object')
LOCATE OBJECT:
        execute and display the syntactical relations between entities of an object
GENERATE FACTS:
        produce semantic relations between objects
INTERPRET:
        check the validity of a design solution against rules in the system's knowledge base
```

Designers make use of all three classes of knowledge and apparently utilize morphisms (different word or inflection forms) between them which change their function. Thus, performance knowledge at one time becomes a goal set, whilst at another time it becomes part of a constraint set. Syntactical knowledge is used both to generate designs and to check generated designs, and so on. Design rules attempt to encode experiential and phenomenological knowledge in a formal and structured way. Knowledge accumulated from design experience sometimes results in heuristics or rules-of-thumb, which enable the designer to focus quickly on important facts of existing conditions, and the match, through knowledge, of the appropriate patterns and elements which fit the needs of the design requirements and the design goals. Such knowledge and experience may be encoded in a production rule-based formalism. The resulting system may then utilize the knowledge of human experts as the basis for design and composition.

Automated design methods are in current existence in a number of structural design domains, for example: high rise buildings (Maher 1984), structural steel connections (Adeli 1988) and steel elements (Ghosh and Kalyanaraman 1993), architectural building eave detailing (Mitchell and Radford 1987), architectural building code checking (Dym *et al.* 1988), and a host of others found in Wexler (1984) and Rychener (1988). Automated design methods specifically for timber connection design in particular are few in number, and are usually connection-type specific (Thomson *et al.* (1987), Neis and Neis (1993), Simpson Strong Tie (1993), Krasojevic and Haller (1994)). An interesting system proposes mechanically fastened joint designs for wood furniture (Nieminen *et al.* 1990). However, in spite of the few connection design systems for timber structures, none attempt to deal with holistic design and in particular, aesthetics or expression, leaving the issue totally at the hands of any designer. Such programs are either typically hierarchically-ordered using high level programming languages or object-oriented techniques, or are simply clever interfaces to connection databases that are either difficult to add to or restrictive in suggesting creative solutions to the designer. Most adopt an analytically diagnostic solution strategy by taking a user-defined assembly of connection components and analyzing it for suitability; as opposed to a generative solution strategy where multiple solution possibilities are generated for the satisfactory joining of two or more abstracted objects such as one horizontal member and one vertical member of given cross-section size and shape lying in a particular orientation in space.

An automated design method that comprises a measure of artificial intelligence is likely to be more successful in suggesting unique connection solutions to the designer. Among the many definitions of artificial intelligence (AI) presented in the literature, the definition this research adopts is:

> a programming style where programs operate on *data* according to *rules* in order to accomplish *goals* - Taylor (1988).

Further to this definition, Born (1987) summarizes the central idea that AI programs should be sophisticated (or smart) programs which allow for (or even themselves take care of) an intelligent use, application or interpretation of computational models (i.e. results), insofar as they are open to revision in accord with new information or insights by the (end-) user. This revision ability is especially important if object-oriented programming languages are used which allow for an immediate evaluation of results and which, in using interface devices such as icons and menus, place the emphasis on the human user and not on the machine. This means that the intelligent use of computers in artificial information processing contains a strong ethical component which includes a scientist's responsibility for the influence of his/her results upon the development of mankind. This caveat regarding the aging of embedded knowledge and principles of truth is also echoed by Wigan (1987) since accepted trends and beliefs have been shown to change with time (particularly in the field of aesthetic styles) which may be valid at the time an expert system was created, but may not be valid in the future. Thus, a facility for ease of revision of a scientist's interpretation of truth is an important consideration in the design and maintenance of an automated system.

However, Lawson (1990) claims that above all, the fundamentally important feature of a computer tool is ease of use. The tool must be intuitive with a flat learning curve so that users will use it with enthusiasm and benefit from its capabilities. Some of the desirable features which make a computer tool a joy to use include:

- a natural communication mode. Human interaction is primarily graphic in nature, and a system interface should reflect this fundamentally natural way of communicating;
- clear instructions and feedback from the system for results and operations to perform;
- quick, intuitive operations and controls for the entry and interaction of data and system controls;
- fast, accurate execution for efficient dependable results of high integrity;
- attractive displays to keep the user stimulated, and for presentation to others; and
- output portability to a variety of media forms or formats for use by other systems.

Thus for a timber connection design system, a graphical interface that works with and manipulates 3D objects directly and explicitly would seem ideal. A windowing system to display graphical and textual information individually or simultaneously would also be appropriate. Each represented connection component in object-oriented programming fashion would include its own comprehensive object attribute list (which may be tied to a database) that accurately reflects current design state object attributes of interest, and object methods which allow the object to respond to any changes in attributes or messages sent to the object. A modular architecture, especially for procedural-type programs, functions or object methods, would provide system flexibility for future updating and amendments. These features among others seem appropriate for the development of a high quality system.

1.2 PROBLEM STATEMENT AND OBJECTIVES

The general focus here is on modular skeletal building systems with a particular focus on heavy timber connections. In this work, research is conducted of current technologies that can be applied to the development of a computer-aided design tool that can be used by architects or engineers for pre-designing timber connections.

The benefits will yield to the designer a significant savings in design time and associated cost, and to the material manufacturer the likelihood of increased specification of product because of a new ease to specify a desired item.

The problem dealt with in this research specifically is: to join members together such that design requirements are satisfied. In dealing with this problem, a good solution should satisfy the following objectives:

- respond to design requirements with a variety of possible connection configuration designs,
- feature a design space description that is simple, modular, easily visualized, and yet can be analyzed to any desired degree,
- easy extension of the system to include more possible design classes, and
- feature a bridge between the numerical world of engineering analysis, and the visual/linguistic descriptive world of architectural design and aesthetics to facilitate holistic design comparisons and evaluations.

Chapter 2 of this thesis deals with the design space description and representing the connection problem in abstract terms, while Chapters 3 and 4 deal with quantitative-qualitative descriptor translation (with two research studies) and aesthetic design issues for connections. Chapter 5 proposes a method to link engineering calculation expertise for detailed analysis into the design process based on the representation method of Chapter 2. Chapter 6 details the computer implementation of many of the ideas of the preceding chapters in the context of a generative design tool for timber connection configurations, supplemented by validation studies of the tool in Chapter 7. Chapter 8 concludes the work here, and suggests future research directions.

Chapter 2 REPRESENTING STRUCTURAL CONNECTIONS

2.1 INTRODUCTION

In a structural building frame, the structural components may include beams, girders, columns, floor, roof and wall systems. Each of the components requires connections with others designed to suit the expected function of the structure, the aesthetic theme of the building if left exposed, as well as provide structural safety with economy. In many cases, the performance of the building is not only governed by the characteristics of the building, but also by the *way* these structural components are brought together. Hence, connections in buildings, and in this thesis timber buildings in particular, are of focal importance in the context of a holistic design process.

This chapter presents a formal representation of connections and a rationalization procedure for design, towards the later development of a knowledge-based design tool for timber connection design. The focus of the problem to be resolved is to join members together such that holistic design requirements are satisfied.

2.2 KNOWLEDGE REPRESENTATION

Connection design can be conceptualized as an arrangement of parts consisting of structural members to be connected, hardware (large intermediate/accessory pieces) to be used in the connection, and fasteners used between the hardware and the structural members. These components of a connection can be viewed as a collection of objects similar to a kit of parts; each part with corresponding functions and behavioural properties. The assemblage of the kit of parts in a holistic context must follow a particular design language such that the *functional* aspects of the components are realized and behave as intended according to the *formal* semantics of the language. If one follows the accepted architectural idiom that *form follows function* (Louis Sullivan according to Holgate 1992); then for structural connections, objects in the connection can be first represented in a general way based on their function. At the functional level, the language can consist of the names of the parts arranged hierarchically. The hierarchical arrangement in itself provides the semantics to the functional language.

The act of assemblage in connection design falls into a class of problems known as configuration problems (Stefik 1993). Much of the design effort in this approach goes into defining and characterizing the set of possible hardware and fasteners for given structural members to be connected. Hardware and fasteners are constrained to come from their corresponding predefined sets. Similarly, the structural members are of another predefined set. The computational model of configuration ensures that the compatibility requirements between structural members, hardware, and fasteners (which form parts of possible connections) are handled in an efficient manner. This is achieved by employing a catalog of parts which specify the requirements that parts have for other parts for their correct functioning or use. The search performed in order to characterize the possible hardware and fastener options is guided by prescribed design requirements for the connection. Design requirements characterize the function of the connection, as well as the safety, aesthetic, and cost requirements.

The configuration model is well suited for timber connection design because hardware, fasteners, and structural members comprising any connection can be selected from three predefined sets. At any given time and location, the sizes of the sets corresponding to possible structural members, hardware, and fasteners, are determined by the timber and engineering wood products manufactured, the hardware and fasteners commonly available on the market, and the preferences and prejudices of the designer. Therefore, it is important that newly-introduced structural members and fasteners to the market can be inserted with ease into their current predefined sets for use in future connection designs.

According to Stefik (1993), the configuration model has four elements:

(1) *specification language* - describes the requirements that a connection configuration must satisfy. For each major function of the connection there is a key component. For example, structural members give load to and take load from the connection. Hardware provides transfer of load between structural members, while fasteners provide for transfer of load between structural members and hardware. Connection design as a

configuration problem requires that detailed specifications of the structural members, hardware, and fasteners be determined for an existing and initially prescribed member that must transfer load through the connection. In the final design, a suitable arrangement of all included parts of the connection must be determined;

(2) *parts submodel* - specifies the kind of parts that can be selected for a configuration and the requirements parts have for other parts. In many cases, the parts submodel may take the appearance of a catalogue. Required parts are given explicitly. The catalogue also indicates which parts are compatible with each other, and what parts can be substituted for each other under particular circumstances;

(3) *arrangement submodel* - describes acceptability and preferences with respect to spatial arrangements or parts. Several criteria are considered in the arrangement submodel: issues related to good engineering practice in timber connection design such as avoiding the possibility of causing tension perpendicular to grain, limiting stresses due to shrinkage, avoiding crushing of wood members due to unconstrained displacement of parts, etc.; and

(4) *sharing submodel* - which describes the load transfer through a connection: through the surfaces of members and hardware. The total load carried by surfaces of a part is distributed by the use of fasteners. Each fastener consumes a certain amount of the hardware surfaces, as well as a certain amount of the surfaces of the connected members. The load carrying capacity of wood members per unit area is required to be compatible with the load distribution capacity of the specific fasteners to be employed in the unit area. For performing the function of load transfer in a connection, certain sizes of shared surfaces are required for both pairs of member and fasteners. Depending on the size of the load to be transferred and the specific members and hardware to be used in the connection, the compatibility of these parts require that the total surface areas needed to be shared between the members and hardware do not exceed the total amount of surfaces available for each part.

The sections to follow describe the configuration model formalism in the context of its four elements above and the underlying process of how connection objects are brought together in the context of holistic connection design.

2.2.1 Physical Objects in Timber Connections

The last three of the four configuration model elements are briefly introduced in this section and are developed more completely through the remainder of the thesis. Timber connection design deals with real life objects or components: members, hardware, fasteners; and a specification language in terms of design requirements and information. According to the most fundamental function of a connection, objects in a joint share the function of transferring load. For example, structural wood members provide the function of giving and taking load from the connection. Hardware, if present in the form of metal plates, hangers or other forms, provides the transfer of load from load Givers to load Takers. Fasteners are used either to distribute the load coming from load Givers to Transferors, as well as to distribute the load from Transferors to load Takers. If hardware is not present, the fasteners may act solely as the Transferor.

The above example illustrates the formation of a continuous load path: from load Giver (member), which may transfer load through load Distributors (fasteners) to a load Transferor (hardware), which subsequently passes the load through load Distributors (fasteners) to a load Taker (member). The assembly of Distributors and Transferor would constitute the physical joint, as shown symbolically in Figure 2.1. A *functional hierarchy* exists as load Giver, load Taker, load Transferor, and load Distributor.

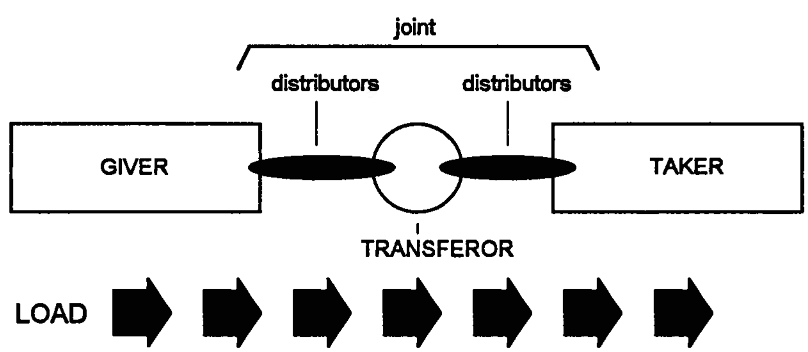


Figure 2.1 Representation of Connection Objects

A Giver or Taker object has the following properties. Because of its typically prismatic (tectonic) nature, the object has side surfaces, two end surfaces, and an axis corresponding to the structure member axis as shown in Figure 2.2. Givers or Takers have material properties (including surface properties), behavioural properties (capacity), and the function of giving or taking load (thus prescribing the load to be transferred).

A Transferor object is an object which is shared jointly by Givers and Takers. Therefore, a Transferor object can be discretized into volumetric portions abstracted as "surfaces" which are identified in contact with the Giver, or Taker, or nothing at all. An example is the Angle object of Figure 2.3. The Transferor object also features an *XYZ* axis system usually oriented with the *Z*-axis aligned with the Giver longitudinal axis, the *X*-axis lying in the horizontal plane perpendicular to the *Z*-axis, and the *Y*-axis lying in the vertical plane perpendicular to the *X*-axis; depending on the Transferor. In all cases, the Giver axis is always aligned with the *Z*-axis.

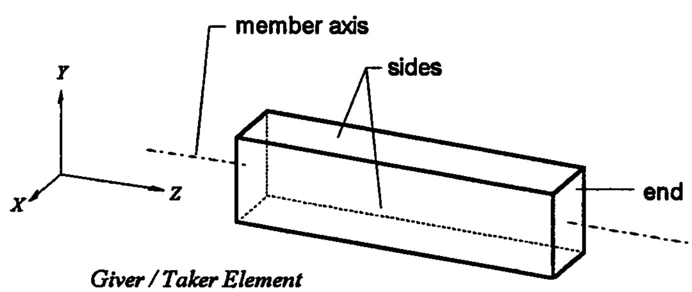


Figure 2.2 Giver/Taker Object

A special surface, known in this study as the *Transfer Surface* represents the common surface between the Giver and Taker where load "transfers" through as seen in Figure 2.3. The *Transfer Surface* is a feature of the Transferor and serves as a reference surface for movement of the Giver or Taker with respect to the Transferor in either the *x, y, z, X, Y,* or *Z* directions, or rotations *R* around the *Z*-axis. The reason for choosing such a directional system is for ease of linguistic reasoning to be performed later and in developing design rules for movement control. The *Transfer Surface* is located in the plane of contact between the Giver and Taker; typically (but not always) parallel to the *XY* plane, and serves to recognize the inter-surface motions which exist at the boundary between the Giver surfaces and Taker surfaces.

Associated with the Transfer Surface, and an attribute of the Transferor is a Restrained Motion Code. The Restrained Motion Code identifies axis directions and one rotation that the Giver or the Taker respectively is prevented from moving with respect to the Transfer Surface. Movement restraint in any particular direction may be offered by the Transferor itself, or by the presence of Distributors that may penetrate or pin a Transferor surface to the Giver or Taker. This approach seeks to address unforeseen movements at a structural connection that could lead to serious problems if the designer is not aware of the possibility of occurrence (i.e. uplift), or redundancy from over-specification of Distributors. The clarity of the workings of the Transfer Surface and Retrained Motion Code will be apparent later in the section on Movement.

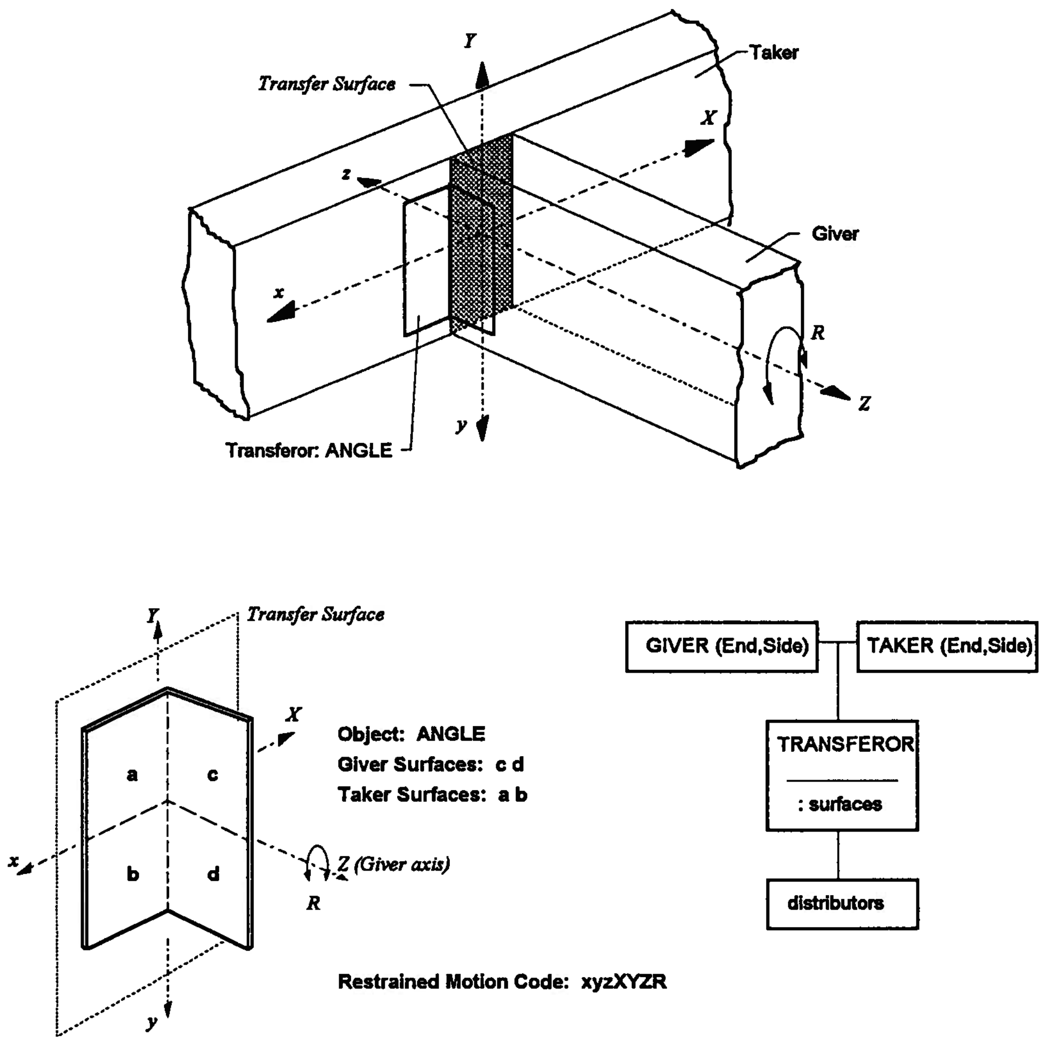


Figure 2.3 Transferor Object

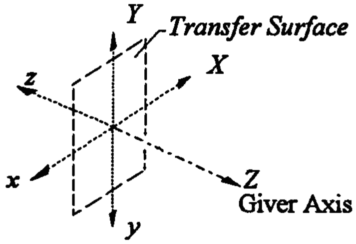
A collection of Transferor objects discretized by abstracted surfaces in the above fashion appears in Figure 2.4. Restrained Movement Codes are also listed for each Giver or Taker contact surface with the Transferor. Restrained Movement Fastened indicates that logical fasteners are present penetrating the Transferor surfaces pinning the Giver or Taker, thus preventing movement. Restrained Movement Unfastened, however, indicates that no logical fasteners are present, and that movement of the Giver or Taker with respect to the Transfer Surface is only provided by the respective movement constraint offered by the Transferor surfaces on the Giver or Taker side of the Transfer Surface, accordingly. Note the orientation of the Transfer Surface in the exceptional case of the Double Angle Hanger used to secure two crossing members. Along with the typical descriptive, physical, and behavioural properties of the Transferor surfaces, each Transferor surface also is endowed with the purpose of selecting possible fasteners (Distributors) to satisfy load transfer.

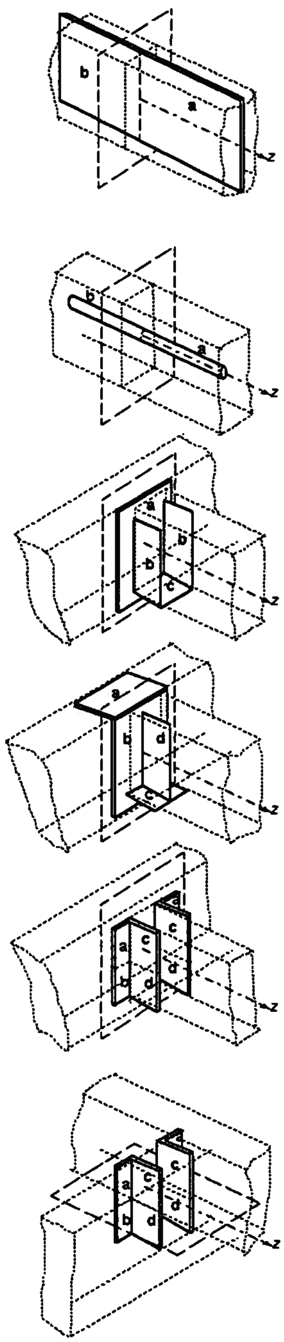
A Distributor object has the properties of capacity (size), appearance (materiality, finish, arrangement, visibility), and the behavioural ability to determine its load capacity.

Basic information for objects and surfaces can be classified and stored in a database for future reference. This feature allows easy expansion by the user without program changes, thus separating program control and the knowledge base.

2.2.2 Specification Language

The first configuration model element is briefly introduced in this section. A specification language for connection design describes the requirements that a connection design must satisfy. An earlier description of this in the context of the configuration model from the point of view of the basic functional connection was offered. However, the traditional players in the design process contribute differently to the development of a holistic specification language for connection design. In architecture, design requirements are primarily visual in the form of an aesthetic style that communicates the connection function; while in engineering, design requirements for satisfactory connection behaviour and economy of means is of interest.





	Transferor Surface	Shared By	Restrained Movement Fastened	Restrained Movement Unfastened
Plate	a	Giver	xyzXYZR	xXR
	b	Taker	xyzXYZR	xXR
Rod	a	Giver	xyzXYZR	xXyYz
	b	Taker	xyzXYZR	xXyYZ
Hanger	a	Taker	xyzXYZR	Z
	b	Giver	xyzXYZR	xXR
	c	Giver	xyzXYZR	y
Concealed Hanger				
	a	Taker	xyzXYZR	Y
	b	Taker	xyzXYZR	Z
	c	Giver	xyzXYZR	y
	d	Giver	xyzXYZR	xXR
Double Angles				
	a	Taker	xyzXYZR	Z
	b	Taker	xyzXYZR	Z
	c	Giver	xyzXYZR	xXR
	d	Giver	xyzXYZR	xXR
Double Angle Hanger				
	a	None		
	b	Taker	xyzXYZR	YZ
	c	Giver	xyzXYZR	xyXR
	d	None		

Figure 2.4 Transferor Object Surface Identification

In the case of development of design requirements for timber structures, designers often recognize that wood has a natural warmth and appearance that begs to be exposed and expressed in a building, and thus timber structures are normally exposed structures which have advantages and disadvantages, for example from Thornton *et al.* (1993):

Matthys Levy, PE, *President*, *Weidlinger Associates, New York*:

> ....I still feel that structural expression is not only pleasing, it's also highly economical. It has secondary benefits. For example, when a structure is exposed, you can always see what's happening to it... if it's behaving well. Though it's certainly not for every building.... (The most technically difficult problem in exposing the structure is) the details and the proportion of the details .... how a particular joint is worked out....how you satisfy the need for two different kinds of joints, some fixed, some sliding. The details are what the structure will be judged by. One thing I thought was interesting recently is the work a Swiss engineer is doing with wood structures. He's been developing a lot of fun joints for wood, very expressive joints for the trusses and beams. I think it has brought back a kind of structural romanticism to designing wood structures......Wood joints are very critical. They are hard to design because wood has characteristics unusual in a structural material. Unlike steel, wood doesn't have the same properties in all directions. Plus you can't weld it, and even when you bolt it, you have to be careful....it's very tricky.

Peter Rice, *Director, Ove Arup & Partners, London*:

> ...I take the view that the use of structure as an exposed element is almost an architectural decision, part of an architectural philosophy that requires a degree of dialogue between engineer and architect. It's not necessarily an extended dialogue, or one that is particularly sophisticated, because sophisticated structural concepts don't usually make good architecture. I have a rough principle that I use in the design of a structure. Never do in one joint what you can do in two. It's difficult to read the performance of a joint that's doing two jobs. Exposed structure as an architectural element is all part of a highly traditional approach to building. It may not be part of the short-term American tradition, but it might be part of the long-term tradition.

According to Carmichael (1984), to appreciate the complication of design issues, design requirements in terms of good timber detailing can mean different things to different people:

- to lay public: questions do not arise until problems occur.
- to the architect: good clean lines, pleasing grain, neat unobtrusive jointing devices, sound weathering details, etc., feature a greater degree of importance.
- to the engineer: requirements for detailing a structural joint, economics of timber species, consideration of chosen size for loads to be carried feature more highly in thinking.

No matter the discipline involved, the important word is *detail*, for in this word, Carmichael claims lies the key to success or failure. This raises many underlying questions in the design process as a whole, such as:

- Have I fully understood the task at hand? If not, think again or seek more experienced advice.
- Is my approach the best solution or are there other ways? If so, I may need to examine more than one path and make comparisons.
- Does the detail I am about to employ satisfy any architectural constraints?
- Am I clear on the general specification and do I comply?
- Are the materials employed readily available? If not, what are the alternatives?
- Does the detail require a factory process and , if so, can it be made? If in doubt, consult the fabricator.
- Does the detail require an in situ application? If so, is it practical? Are their any site obstructions to consider? Does it require special tools? If in doubt, consult the builder.
- Have I given all relevant information to all concerned?
- Finally, but equally as important as all the previous questions, have I given my client what was requested, employing safety and economy efficiently and to the best of my ability or could I have done better?

For timber connections, guidelines on aesthetic expression are extremely rare and are only briefly cited (Canadian Wood Council 1991, Stalnaker and Harris 1989, Goetz *et al.* 1989). The rarity of guidelines is one of the reasons why engineers often give this matter little thought. However, these guidelines often greatly simplify the need for a more rigorous analysis using formal aesthetics as described by Edel (1967) and need only to be applied in design. The most common aesthetic issue for mechanical timber connections is visibility in terms of scale, and blend/contrast with the connecting members that in combination leads to an appropriate desired expression of the architecture. For example, if a clean uncluttered appearance is sought in order to express the qualities of the wood members, then hardware and fasteners are often hidden, or fasteners of very small visible head sizes are chosen to accomplish this goal. The key word here is *clean* which in itself represents the desired expression in the aesthetic specification language. Further treatment of this issue appears in Chapter 4.

Simplified techniques based on rules of thumb for the structural design of bending members exist in Taylor (1982), however the same kind of knowledge for connection capacity has not been well documented in the literature. Detailed engineering analysis techniques for timber connection behaviour with any real reliable precision is also lacking as reviewed by Taylor (1991), partially because of the acknowledged lack of complete understanding by researchers of connection component behaviour. Simplified conservative techniques, when available, are often tied to a particular piece of hardware or fastener. Most often, practical rules of thumb are followed for initial selection and configuration of connection hardware and fasteners, which is subsequently verified using design code rules for capacity. This is discussed further in Chapter 5. The engineering design result may have implications on the desired architectural expression that may require attention. Thus, a balance must be struck between architectural concerns and engineering concerns to render a solution where the structure is expressed in the architecture.

Hence, connections in timber buildings have a demanding specification language in that they are difficult and very time consuming to design well in a holistic way and therefore presents an excellent challenge for implementing an expert system approach to architectural/engineering design.

2.2.3 The Design Process

The question arises as to how the concept of load transfer fits into the design process. Figure 2.5 adapted from Mitchell (1990), Mitchell *et al.* (1991), and Oxman and Gero (1987) shows that the design process proceeds from

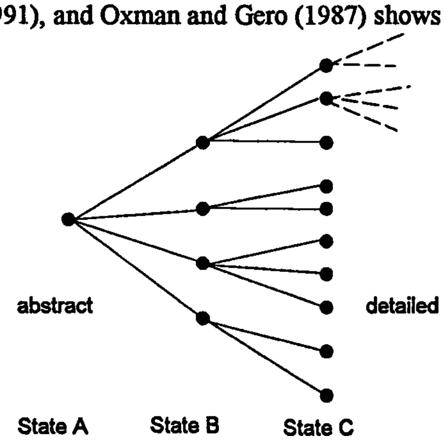


Figure 2.5 Design States in the Design Process

a point of abstraction, through a series of design states, to arrive at suggested solutions (detailed) that are consistent with design needs. The initial design state is typically knowledge of the Giver and Taker (abstract form), plus design needs (design criteria, etc.). The final design state consists of the initial design state, plus additional hardware and fasteners, plus arrangement of all parts satisfying design requirements. The design process then can be modelled as shown in Figure 2.6. The dashed line refers to direct user assessment and action, while the remaining steps have significant automation potential. The automation potential can range in degree from total automation to partial automation (user-interactive by automation over-ride).

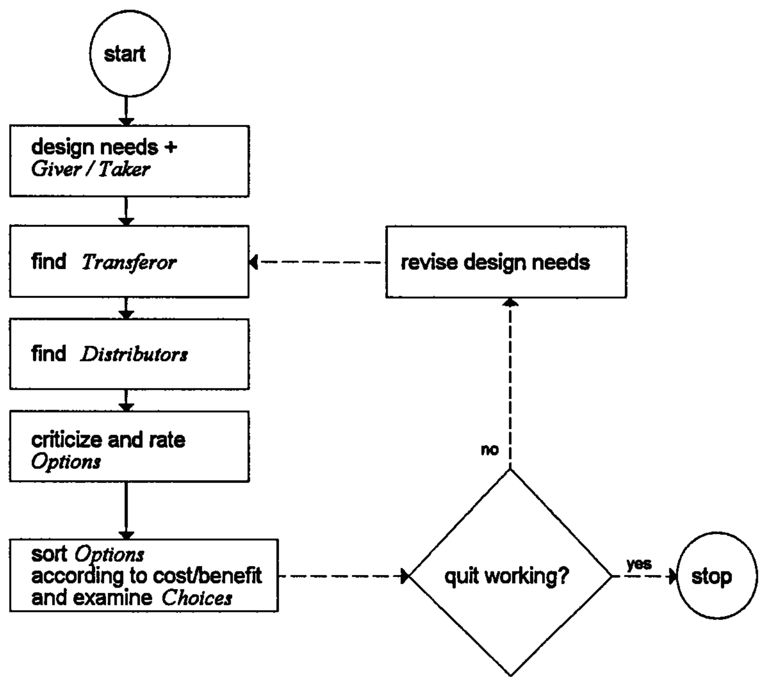


Figure 2.6 Connection Design Process

In the automated mode, knowledge of the Giver and Taker identifies a list of Transferor objects which meets criteria for member contact planes (compatibility), and design needs (visibility, preference, fire protection). Each Transferor object in the list determines its own size, material features, cost, and surface characteristics based on its own behavioural functions. Each Transferor surface identifies a corresponding list of Distributor objects which meets criteria for design needs (visibility, preference, appearance, fire protection). Each Distributor object in the list also determines its own size, quantity, material features, and cost based on its behavioural functions. Finally, each unique combination of Giver, Taker, Transferor, Distributors yields a joint design Option.

Each Option can now be criticized and rated according to the following criteria: stability of members in the joint (shrinkage, uncontrolled movement), capacity over/under design, fire protection, appearance (aesthetics, expression), and cost performance. The rated Options are then ranked based on best cost, best performance, or best overall, and offered as a list of solution suggestions for the user to explore. The performance rating is simply the sum of ratings from design criticisms (non-weighted). If none of the solutions is agreeable with the user, then the user must alter the initial design requirements accordingly.

2.3 RELATIONSHIPS AND REASONING

As a brief introduction, some general background found in the literature on relationships and reasoning in the design process is presented, followed by a focused discussion on connections in structural timber. The relationships and reasoning within and among the objects in the design process typically deal with either design concerns or physical behaviour. Akin (1986) describes some formal reasoning and search methods in this context as, in the global sense in terms of the total solution strategy:

- *depth first*: allocation of the designer's attention to the siblings of a parent node before moving to the next parent node of the same depth in a tree-like search space. In spite of the method's popularity, it does not guarantee optimal search of the tree, does not insure finding a solution, and does not insure a balance of emphasis between multiple issues of equal priority that may be implied by the levels of the tree structure,
- *breath first*: the designer studies each problem component in a lateral sequence, not allowing the opportunity to search any one component in great depth before identifying a solution type and constraining the problem further. The main advantage over depth-first is largely a result of greater likelihood of finding a solution in a shorter time, especially when a large repertoire of prototypical solutions are available,
- *sacrificing solutions*: from a handful of alternative initial solutions, one alternative is examined in detail and the selection made is never subsequently reversed;

and in the local sense as:

- *generate-and-test*: a process consisting of projecting information (generate a partial solution), confirming the information (test if the information meets a design goal), and represent the information in the form of the solution if goal is achieved,
- *hill-climbing*: a variation of generate-and-test where one acquires information (the best-so-far solution), projects information (generate a new solution), confirm the information (compare partial solution with best-so-far solution) and represent the information as the new solution if better,
- *heuristic search (means-end-analysis)*: a process consisting of regulating control (selecting a heuristic method), representing information (the partial solution), confirming the information (does solution meet goal?) and projection of information (applying the method).
- *induction*: a weak method of problem solving consisting of acquisition of information (selecting rules that match predicate), projecting new information, confirming information (does new information match desired goal?), and representing information.

Reasoning through a knowledge tree can be performed in a top-down fashion as in forward-chaining (general to particular), or bottom-up as in backward chaining (particular to general). Reasoning in nature can be monotonic in a sense that a conclusion once found to be true, remains so despite the addition of any further knowledge. However, according to Sham (1993) and others, design reasoning is often nonmonotonic in the sense that new knowledge can change existing knowledge and any deductions made from it (characteristic of inference based on conjecture and insufficient information - an appropriate logic for supporting design). Hence, inferences made by the system are refutable and have to be modified in light of circumstances. Some of the causes leading to nonmonotonicity cited by Sham (1993) are:

1. decision making using incomplete information and a vague design brief,
2. generation and revision of design assumptions,
3. change of design intentions or requirements,
4. relaxation or intensification of design constraints, and
5. detection of contradictions in the decisions made.

Rule-based systems have been used in design systems in the past. This approach uses predicate logic forms to establish relationships between facts in the knowledge base, stimulates experimentation, and is a natural device to express the rules of the game played by designers. Through their flexibility and modularity, rule-based systems offer a potentially very exciting medium for the exploring that characterizes the core of architectural design. The ideal system characteristics:

1. enable designers to simplify contexts and rules in an easy way using graphical means as much as possible,
2. show designers the various ways in which rules can be applied, and
3. make it easy for designers to modify rules and edit the evolving design.

Relationships in the rules can be constructed using mathematical formulae, or natural linguistics as the case demands. In some cases, a simple non-weighted rating value is applied on the outcome of a rule and is used as a measure of the severity of a deficiency; and later, on summation, as a measure of the quality of the joint overall. The formation of these relationships is presented below.

2.3.1 Object Attributes

In some cases, it is possible to assert other parameters about an object based on existing numerical, geometric, or linguistic parameters. Rules can be constructed to infer certain facts about the object based on these existing attributes. This becomes particularly important, as will be seen later, in providing a more complete description of the object, not just in numerical or geometric terms, but in linguistic semantic terms so that other inter-object relationships can come to bear. An example of this is the materiality of an object. A number of other facts can be construed based simply on the material from which the object is made. These facts can be relative, numerical, or emotionally evocative in nature. Table 2.1 tabulates rules for inferred facts from object materiality for a number of object materials. Subjectivity, as will be discussed more fully in Chapters 3 and 4, can play a role in some of these inferred object facts.

Table 2.1 Inferred Facts from Object Materiality for Various Materials

Attribute				Material			
	Steel	Plastic	Aluminum	Concrete	Engineered Wood	Natural Wood	Glass
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Elasticity	Elastic	Elastic	Elastic	Brittle	Brittle	Brittle	Brittle
Hardness	Hard	Moderate	Moderate	Hard	Moderate	Moderate	Hard
Origin	Synthetic	Synthetic	Synthetic	Natural derivative	Natural derivative	Natural	Synthetic
Strength	High	Low	High	High	High	Low	Low
Temperature	Cold	Cool	Cold	Cold	Warm	Warm	Cool
Weight	Heavy	Light	Light	Heavy	Moderate	Moderate	Light
Initial MC	Dry	Dry	Dry	Slight	Dry	Green	Dry

If <MATERIAL> Then <ATTRIBUTE>

2.3.2 Compatibility

Compatibility deals with the fit of the parts in the assembly, and the knowledge that is used here is common sense. Previously, all the surfaces of the Giver, Taker, Transferor have been identified and stored in a database, so the matter becomes one of simply matching corresponding surfaces between Giver-Transferor, and Transferor-Taker. Using the properties of a tectonic element as shown in Figure 2.2 as having ends and sides, a simple set of connectivity rules can be constructed. A simple rule example would be:

IF Giver = End and Taker = Side and Transferor = EndSide THEN Select Transferor as a candidate
ELSE Reject Transferor as a candidate

A more complete logic flow for compatibility issues is shown in the flowchart of Figure 2.7.

2.3.3 Fire

The ability to deal with fire is a major concern in joints of timber structures. In some ways, particularly in larger sections, natural timber performs better in fire than unprotected steel because of its charring characteristic. Unprotected metal in a fire has a tendency to become highly plastic with high temperatures and dramatically lose its strength early in a fire event. Fire protection of timber joints usually amounts to careful detailing to insulate metal fasteners and hardware from the heat of fire by either embedding fasteners and hardware to at least the depth as that required for the sacrificial timber and suitably fire stopping all holes using glued wood plugging, or covering the exposed fasteners with a suitable fire resisting material which gives the notional fire period required. Exposed nails, screws, or staples may be used to fix this protecting material but special attention should be paid to the detail to ensure that the material remains in place for the required period of fire resistance. From these two choices, typically it is prudent to use the embedment method wherever possible because, although more expensive, the approach is more positive in solving the problem and gives a cleaner appearance.

The objective of this module is to provide a means of dealing with fire resistivity of the designed connection. Models to calculate fire resistance rating of wood assemblies have been developed that predict char depth, temperature distribution in the unburned part of the member, and the strength properties of wood at elevated temperatures (White 1988). Fire resistance rating equations for structural timber members are given in most code references (Canadian Wood Council 1990, Goetz *et al.* 1989, Stalnaker and Harris 1989). The analytical nature of these models is suited to development using a high level language or spreadsheet approach. However, fire resistance is a user-specified design requirement that basically identifies if fire resistance is required or not. If not, fire requirements have no impact on the design solution suggestions. However, if fire resistance is an issue, then Transferors are required to be covered with insulating material, or wood, regardless of the visibility requirement. The same rule holds true for Distributors. The intent of these simple rules are consistently found in the mentioned references and thus provides a non-analytical means to assessing the fire resistance of a joint configuration in a preliminary way. Figure 2.8 shows the flowcharted logic for these rules.

2.3.4 Movement

Uncontrolled member movement can result when an unexpected force acts upon an unsecured member, or when its cross-section depth to width (aspect) ratio becomes too large when only the section width is supported. As part of the surfaces database, information is stored about the restraining movement ability of the Transferor if a particular surface is fastened or unfastened to a Giver or Taker. Possible motions include all or some of: x or X, y or Y, z or Z, or R, according to the Transfer Surface reference axes, as seen in Figure 2.3. Using a technique

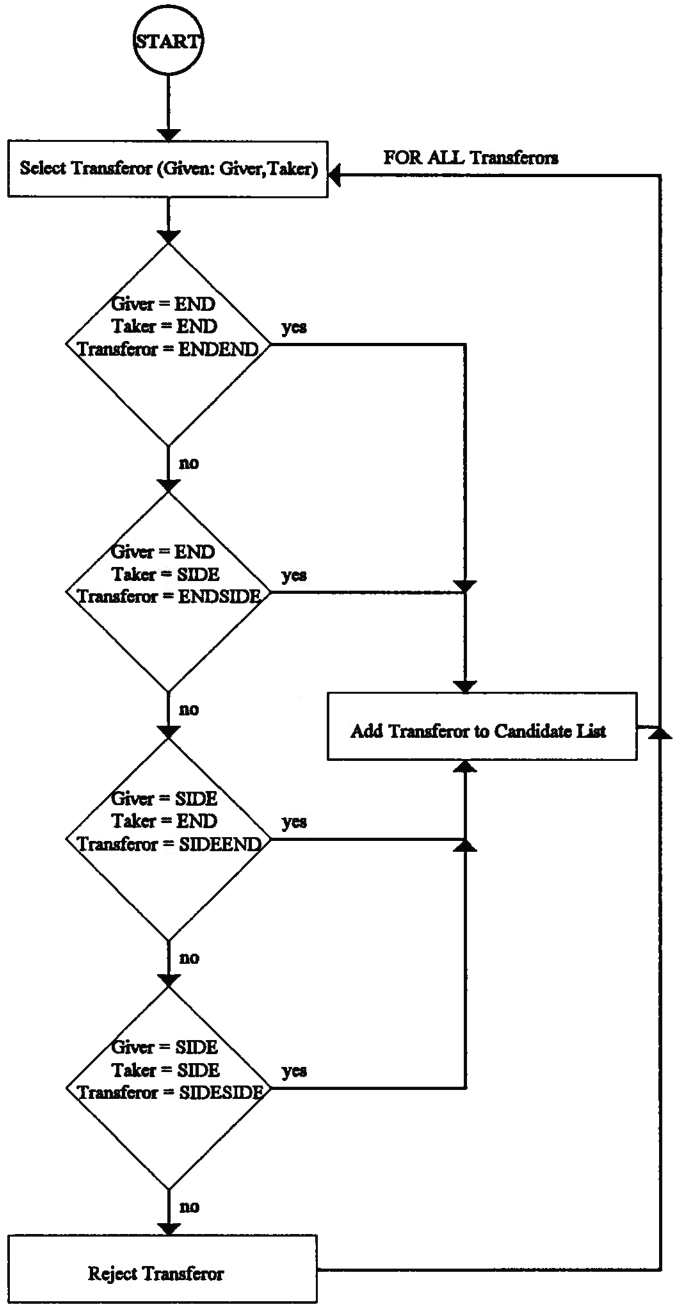


Figure 2.7 Logic Flow for Transferor Compatibility

known as *pattern matching*, the rule simply stated from common sense is:

IF all Transferor Giver-surfaces produce a restrained movement code containing at least one character each of x, X, y, Y, z, Z, and R,
THEN the Giver member is secure in the Transferor and will not move relative to it.

If one of the restrained movement code characters is missing, then motion will be possible in the missing character direction. The same rules can be applied to the Transferor Taker-surfaces to assess Taker movement, and the flowcharted logic for movement appears in Figure 2.9.

2.3.5 Shrinkage

Shrinkage/expansion of wood members in joints subjected to variable humidity service conditions is relatively unique to timber design, and an issue of concern as identified by Goetz (1989), Canadian Wood Council (1991), and others. Relationships for the response of wood to varying humidity and temperature environmental conditions are available in Forest Products Laboratory (1987). However, humidity service condition is a user-specified design requirement that can more simply be assigned a value of constantly dry, variable humidity, or constantly wet service environment. Rules from simple logic are checked according to the decision tree in Figure 2.10 to see if wood members will respond unfavourably in the connection design. A demerit rating is assigned increasing in value from zero depending on the seriousness of the incompatibility. This demerit rating value is later used in criticism of the design against other design candidates.

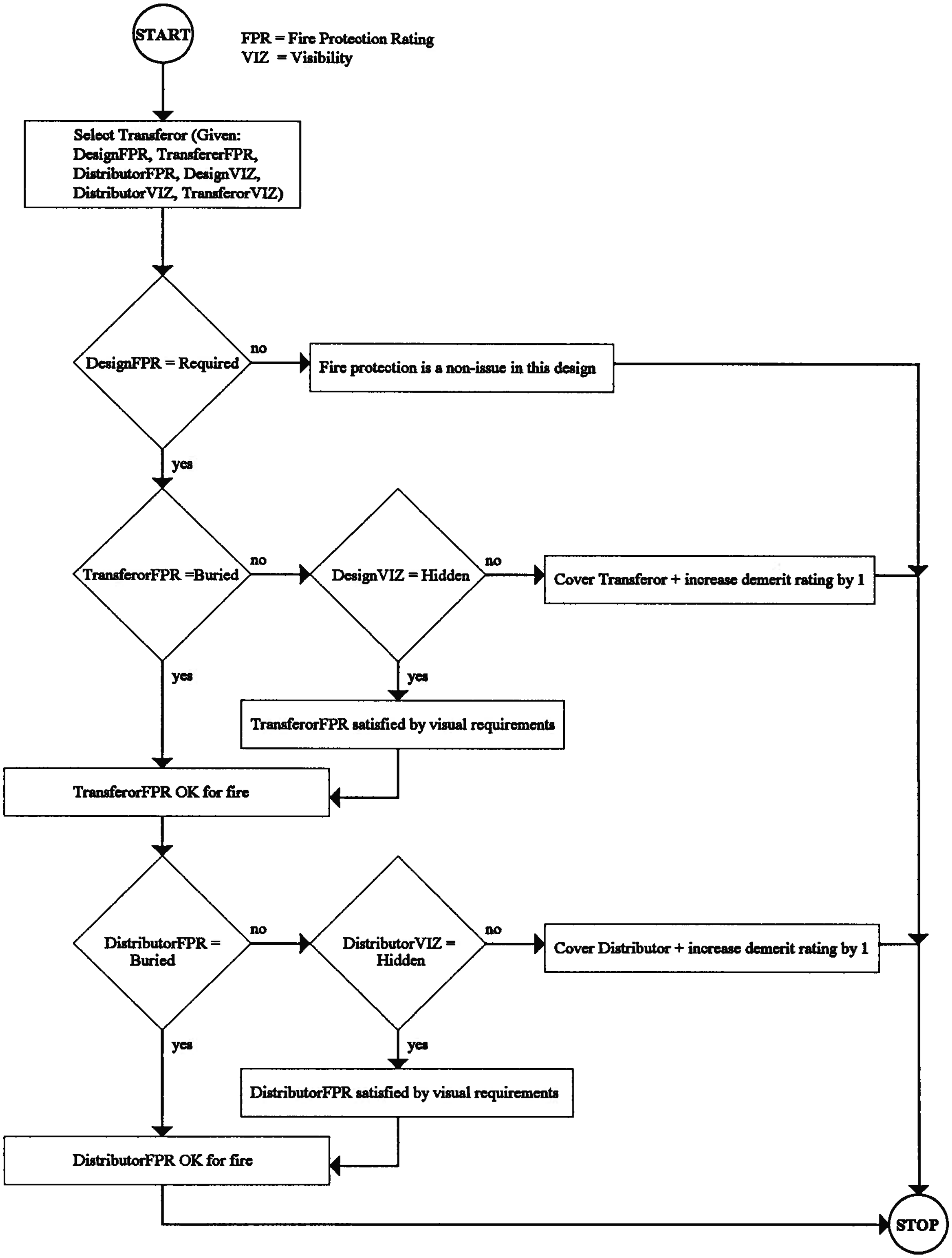


Figure 2.8 Logic Flow for Fire Evaluation

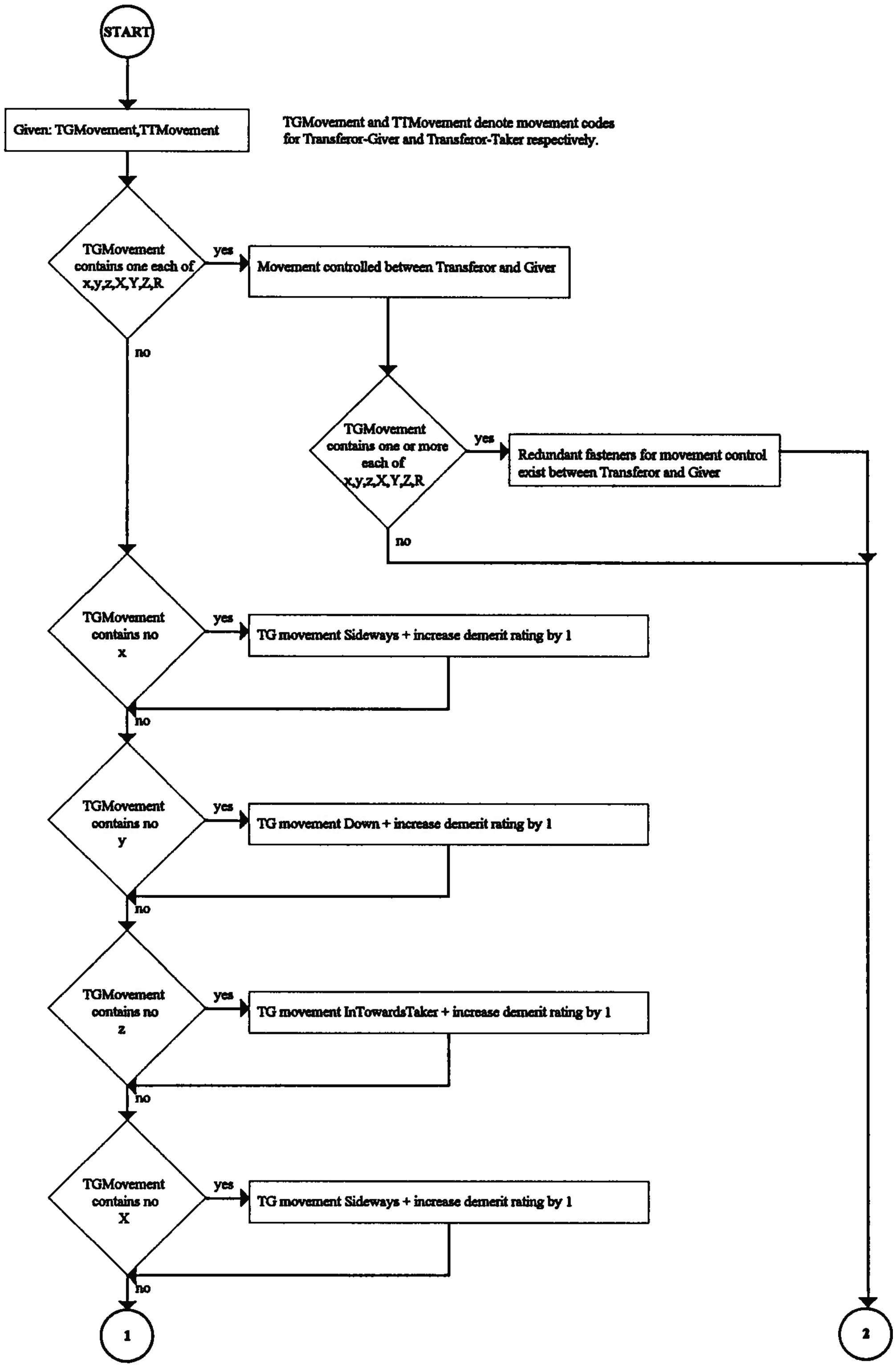


Figure 2.9 Logic Flow for Member Movement with respect to Transferor

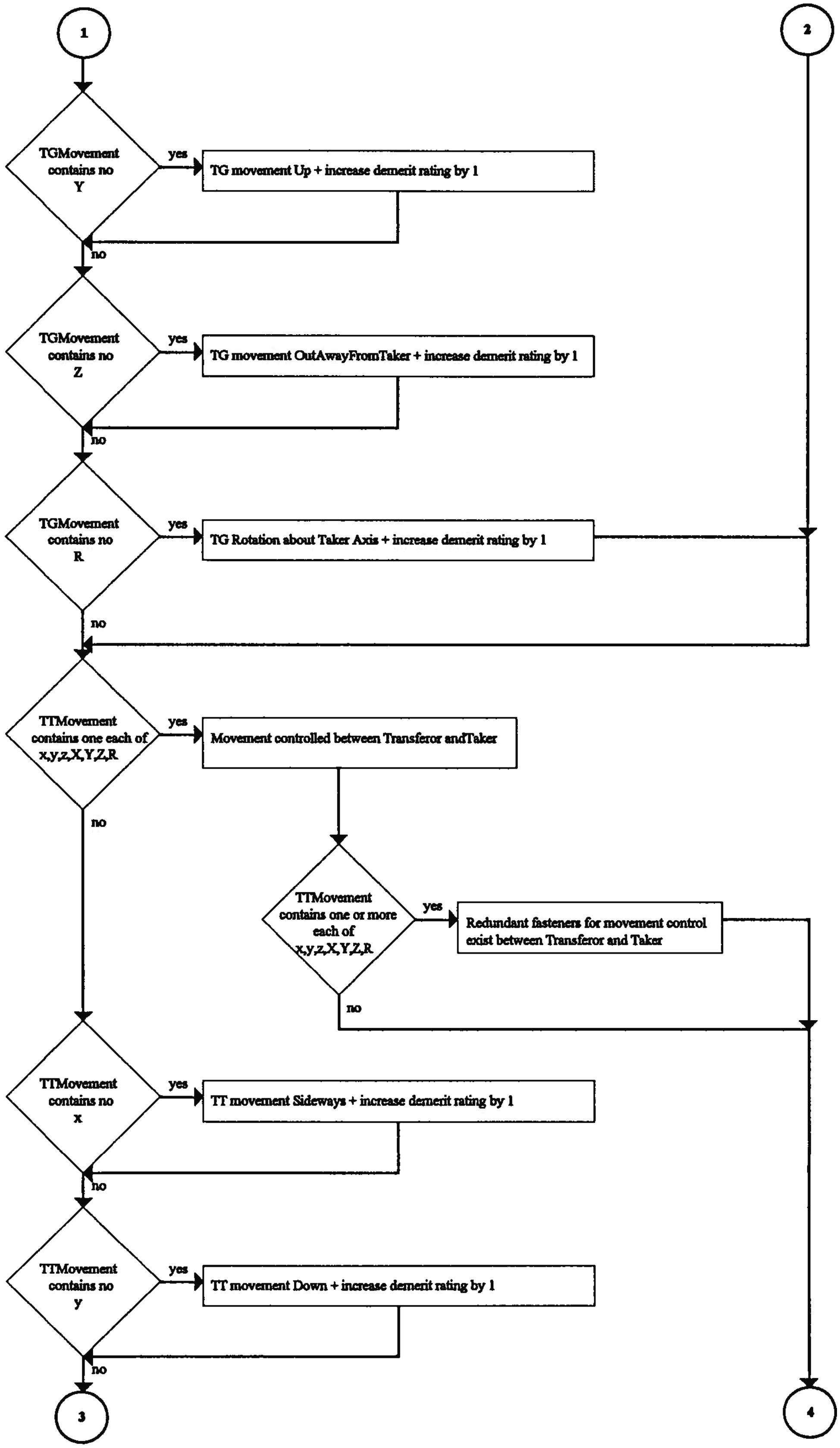


Figure 2.9 *continued*

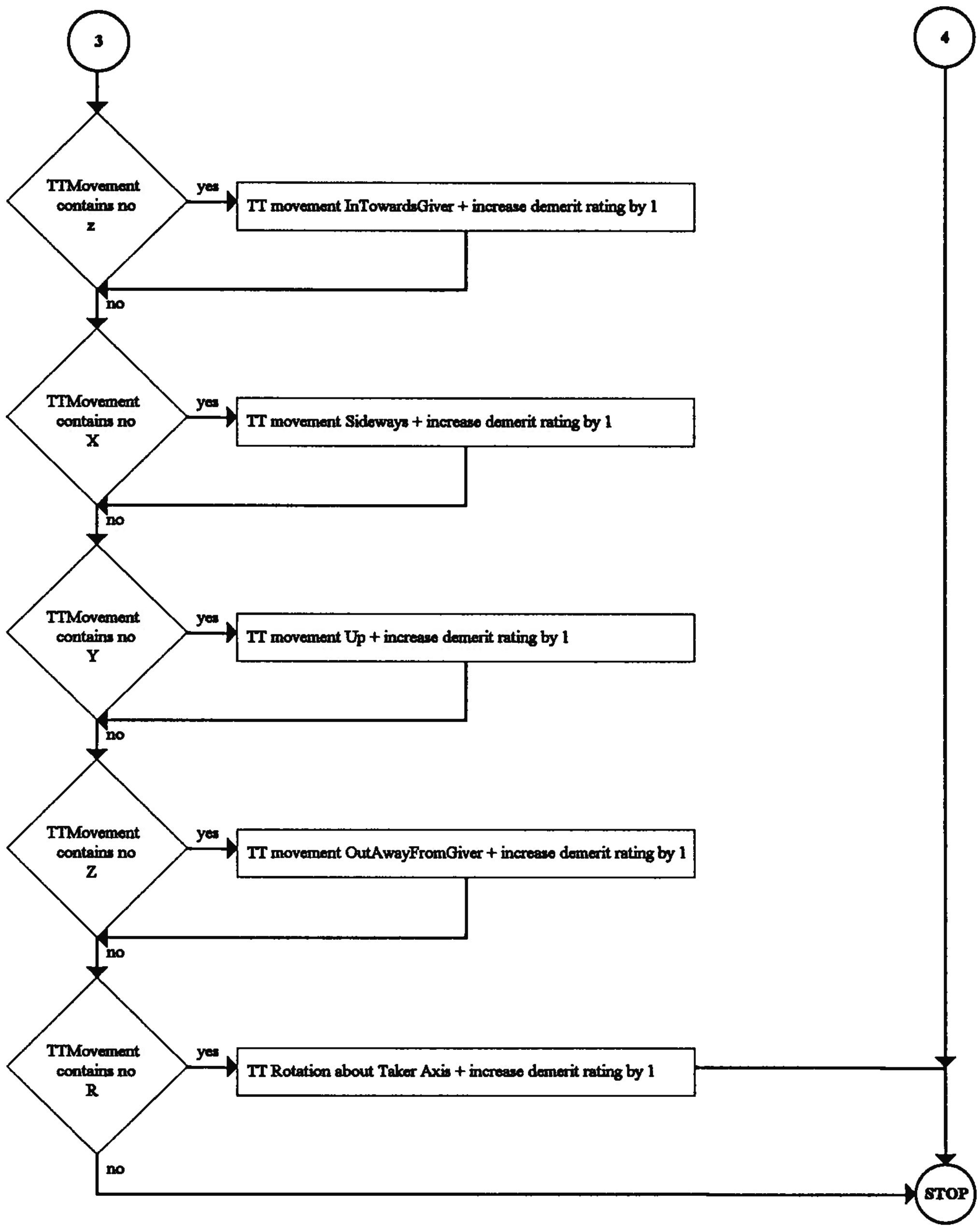


Figure 2.9 *continued*

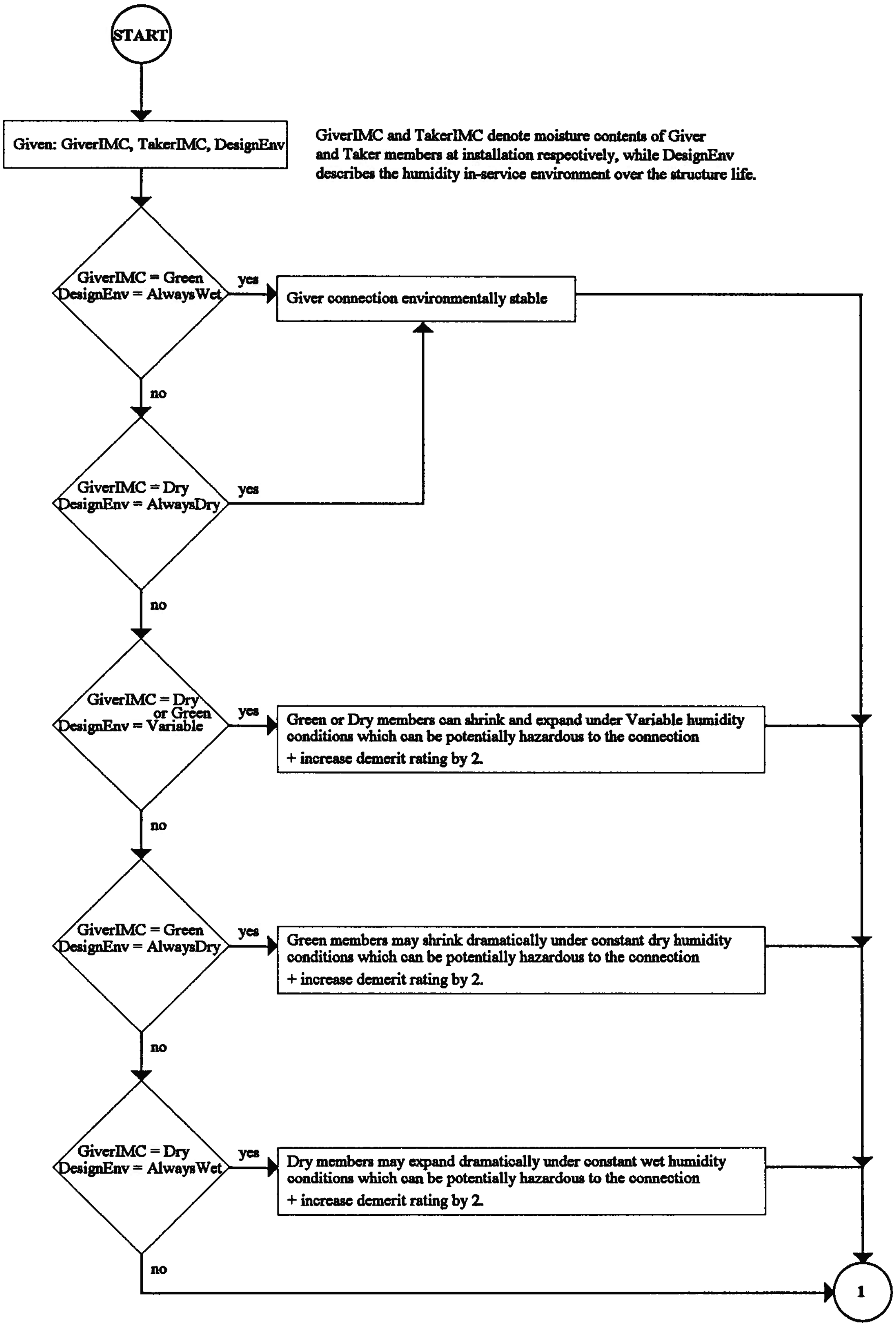


Figure 2.10 Logic Flow for Member Movement with respect to Environmental Humidity Conditions

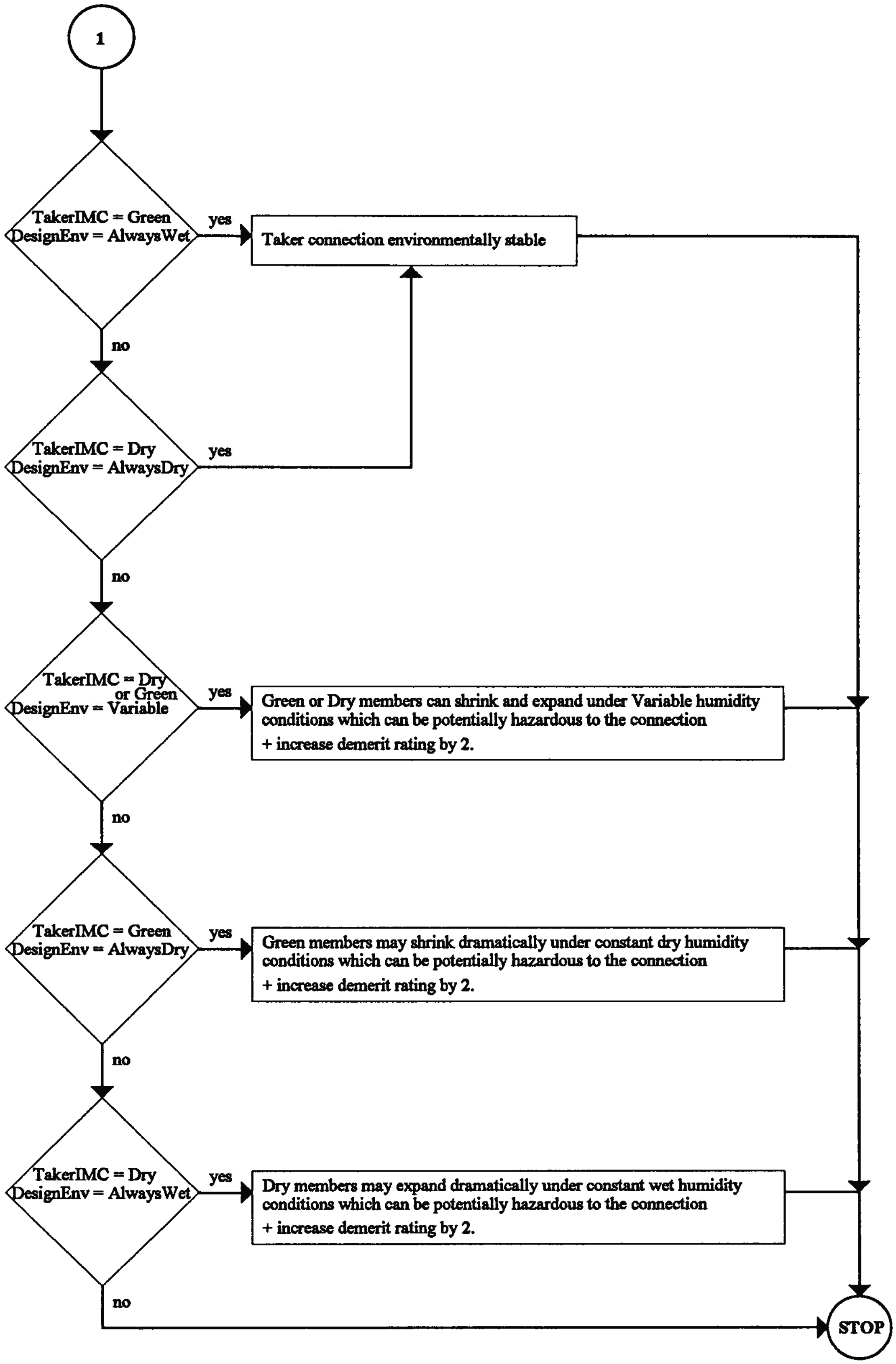


Figure 2.10 *continued*

2.3.6 Aesthetics

Consideration of aesthetic issues seeks to represent the symbolic and formal aspects of an architectural work in a new and novel approach that builds on, without stifling, the intuitive or creative capabilities of the architect. A physical object has formal attributes that can be readily quantified in syntactic terms; attributes which, when combined in a certain way, embody a symbolic/semantic meaning or representative name which can be assigned consistent with external contextual criteria. Hence each object description has a semantic attribute in addition to its physical syntactic formal attributes. When a collection of objects with respective semantic attributes are grouped together, an overall semantic description for the group can be assigned. This idea embodies the architectonic principle from Shack (1989) that the part serves to inform the whole insofar as the whole serves to inform the part, so that design intentions as built order are realized as an architectural experience. Vallhonrat (1988) states:

> Tectonics depends upon very few fundamental aspects of the physical world. One, of course, is gravity and the physics that goes with it. Gravity affects what we build and the ground beneath it. Another aspect is the structure of the materials we have, or make; and a third is the way we put those materials together. How and why we do it affects the way they appear as the surfaces that bound space.

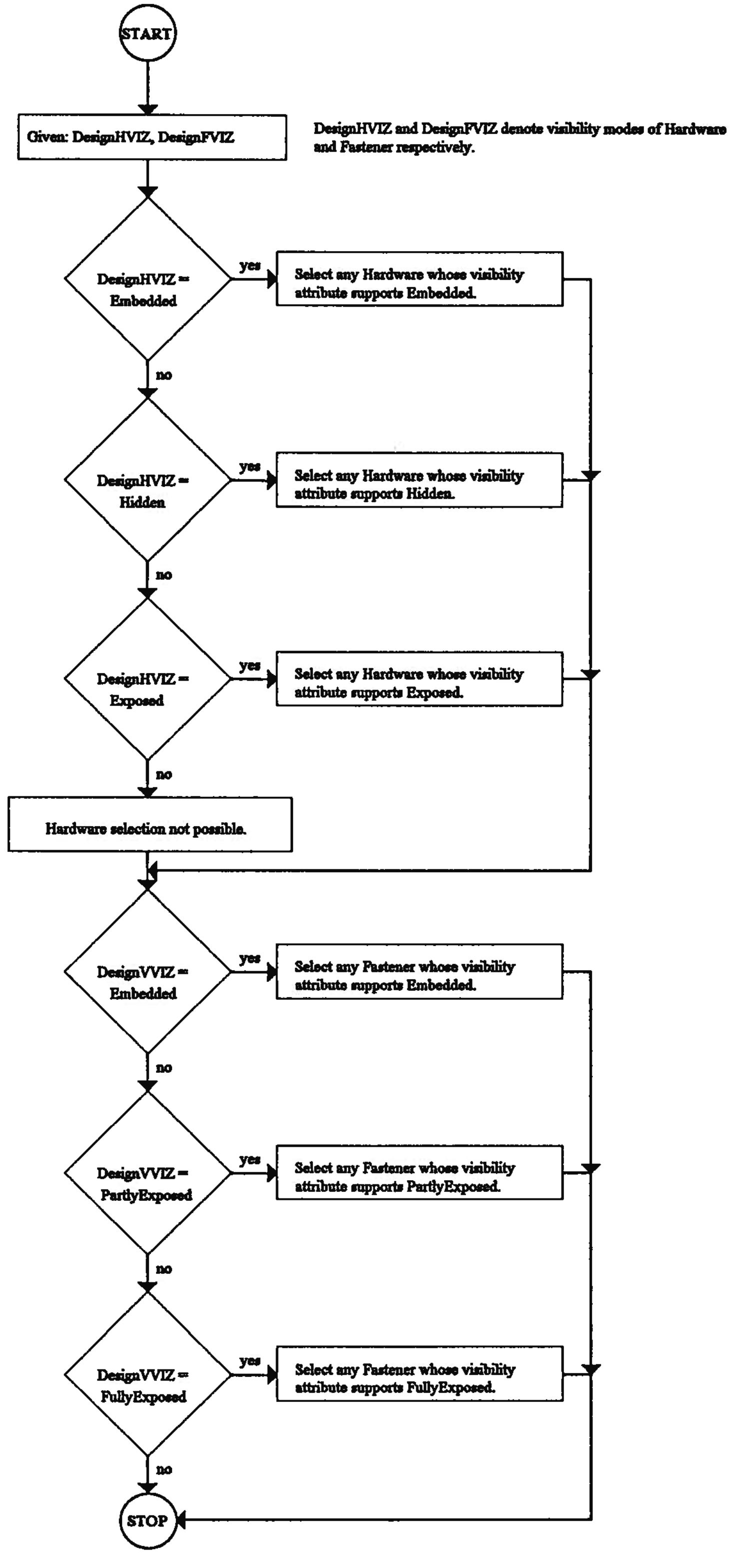
The designer is responsible through his own training and creative talents for developing and assigning the semantic design information consistent with his design intentions. According to Edel (1967) and Holgate (1986), aesthetics of an object or group of objects can possess any of the aesthetic attributes: dominance, rhythm, balance, variety, contrast, transition, and unity. For an artifact to succeed aesthetically, strong representation from up to three aesthetic characteristics must be present in the artifact. If the aesthetic information is captured in a "group semantic / object name / object semantic / object formal description" - structured data base, then a library of aesthetic references is available for future designs, or other designers. The ability for another designer to tune the aesthetic data base to his particular design style or semantic vocabulary is an added exploratory benefit. In a sense, this approach allows the designer a degree of creative aesthetic latitude consistent with the explorative nature of architecture design in a range from the proven to the completely new, while building new aesthetic design rules in the process.

The above may be true for world objects in general, but the intent of the ideas can be greatly simplified for application to timber connection design. Distilling guidelines on aesthetic expression for timber connections cited in Canadian Wood Council (1991), Stalnaker and Harris (1989), Thornton *et al.* (1993), and Goetz *et al.* (1989) leads to simple rules for visibility in terms of scale, and blend/contrast with the connecting members that, in combination, leads to an appropriate desired expression of the architecture. For example, if a clean uncluttered appearance is sought in order to express the qualities of the wood members, then hardware and fasteners are often hidden or fasteners of very small visible head sizes are chosen to accomplish this goal. The key word here is *clean*, which in itself represents the desired expression in the aesthetic specification language. Further discussion of this appears in Chapter 4.

More often than not, these descriptive design key words have meanings that are quantifiably vague in that direct qualitative-quantitative associations cannot be made. Some device is required to bridge the translation gap, especially when one considers that computers are numerically (quantitatively) based, while human aesthetic reasoning is often linguistically (qualitatively) based. This is an interface problem that will be dealt with in Chapter 3, but for now, Figure 2.11 flowcharts typical logic for some simple aesthetic rules for timber connection design.

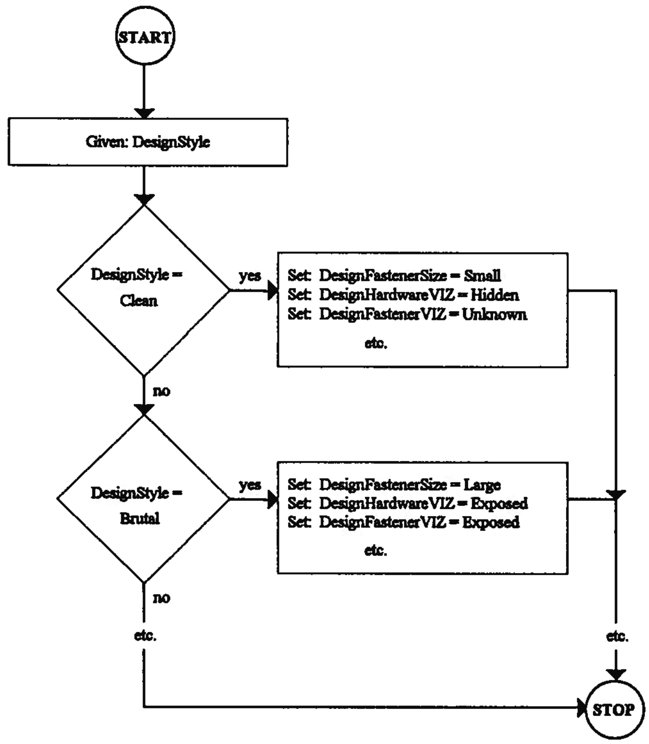
2.3.7 Capacity

The strength design method of timber joints using common fasteners, such as: nails, bolts, split-rings, and shear plates, and fastener hardware, is commonly split into the static analysis of fastener forces caused by external joint moments, shears and axial forces; and, the checking of fastener capacities. Fastener capacities can be determined using the appropriate procedures in CAN/CSA O86.1-M89 Engineering Design in Wood (Limit States Design) (Canada Wood Council 1990) for each fastener type considered. Fastener location and spacing criteria is normally also checked. For steel hardware capacity, appropriate bearing and connection criteria from CAN/CSA S16.1-M89 Limit States Design of Steel Structures (Canada Institute of Steel Structures 1992) is normally followed. The structure of this module is basically algorithmic which can suit development using a high level language such as *C*, *C++*, *Pascal*, *Basic*, or *Fortran*. A modular development architecture within the module

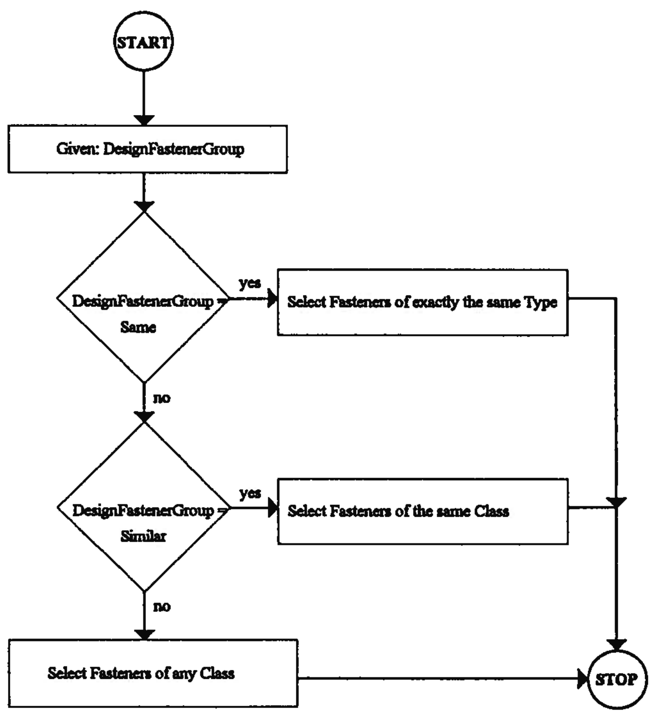


visibility

Figure 2.11 Logic Flow for Various Aesthetic Concerns in Timber Connection Design

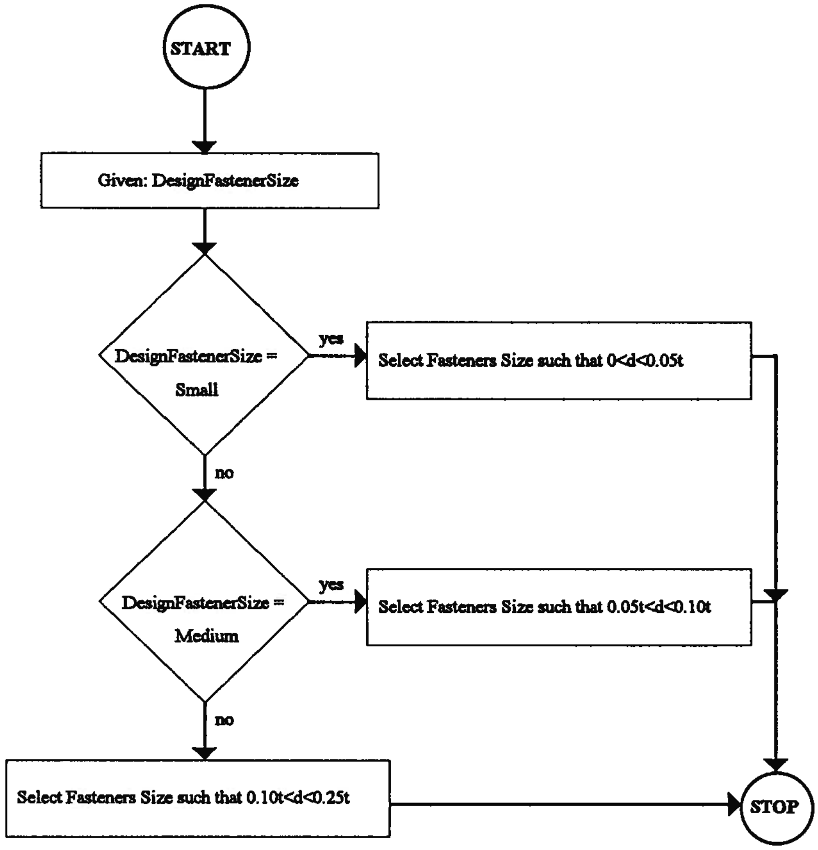


pre-defined design styles by user



fastener type

Figure 2.11 *continued*



fastener size

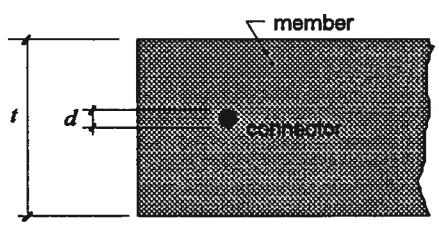


Figure 2.11 *continued*

can facilitate future post-research code upgrades, or a statics engine upgrade.

Capacity of individual connection objects can be determined, as mentioned, on the basis of engineering codes or formulae based on their load resistance according to Limit States design procedures. Applied loads are assigned to surfaces as part of the surface attributes of the Transferor objects in the connection. Simple quantities are determined simply by dividing connector capacities into the applied surface load. Since connection capacity design is an iterative problem requiring detailed knowledge of connector placement, this detail level would require the use of a Transferor-classed analysis engine to handle the force-equilibrium determination for the entire joint. Placement of connectors are governed by clearance rules for ease-of-assembly and code specifications, and rules from design requirements for proximity, grouping, and pattern. An assessment of over/under capacity can be made by dividing the required load by component resistance for the worst connector, and the worst surface respectively. If under-design (> 1), the Option is deleted; if over-design (<= 1), the Option incurs a rating in proportion to the amount of over-design.

A pilot study, TimberCon, detailed in Chapter 5, was conducted to demonstrate the feasibility of such a scheme, and used a spreadsheet approach (Stiemer and Lo 1988) as the analysis engine. The study showed that coupling object information with an external analysis engine is a viable technique. However, iteration requirements proved the spreadsheet to be too slow as an analysis engine here, suggesting that simple high-level language programs could provide a better and more elegant approach that handles iteration well and avoids the high overhead of launching a spreadsheet program. Each analysis engine name is specified as calling value in the respective Transferor class.

In order to determine a reasonable initial configuration for the connection objects, a simplification can be made. Consider that each Distributor in the joint occupies a prescribed surface area as defined by clearance rules, and that the size of the area is a function of the Distributor's shank diameter. Further, each Distributor has finite capacities in shear and tension that can also be functions of the Distributor's shank diameter. The Distributor shank diameter is a key parameter as many capacity and aesthetic quantities can be determined from it. Moreover, if one considers a volume projecting from the surface area around the Distributor as in Figure 2.12, then installation considerations can be assessed as a minimum amount of clearance in the area of the connector must be present to facilitate the connector's installation. From knowledge of the distributor capacity and zone area, the number of distributors can be determined by dividing the applied surface force by the distributor capacity. The required size of the distributor surface can be calculated by the multiple of the distributor's capacity zone area and the number of distributors. Thus, the size of the distributor surface and number of connectors required for a rough strength estimate for the connection can be obtained. This concept is presented here for consideration but was not developed further in the work to follow.

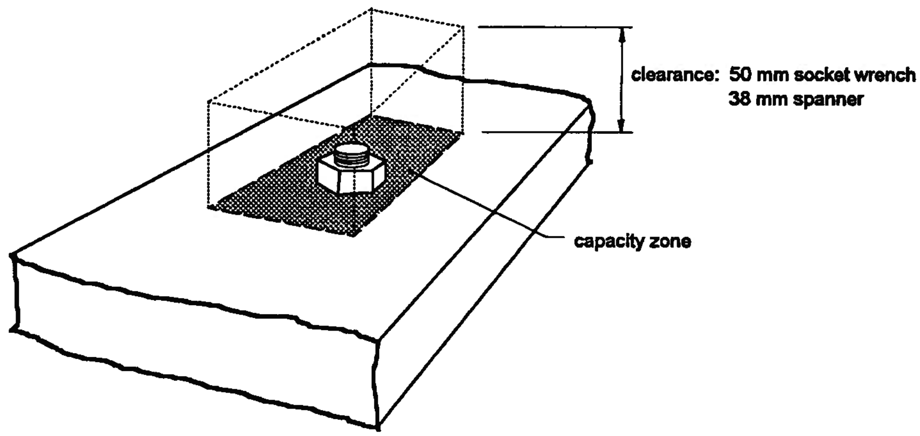


Figure 2.12 Distributor Capacity Zone

2.3.8 Cost/Performance

Economy of means in design is something that is influenced greatly by decisions made in the preceding modules and should be of some concern to both the engineer and the architect. Here, trade-off between qualitative and quantitative issues can be explored. There is often a tug-of-war between architectural intentions and engineering economy of performance which sometimes results in design conflict and professional misunderstanding. While there are no simple answers to this problem, creation of an awareness of the problem constituents is good step, particularly if client money is involved. The objective of this module is to quantify in monetary terms the initial capital cost of the design, and to express it in terms of ratios of engineering / architectural constituents. The cost of the engineering portion is representative of the structural system necessary only to perform its intended structural function for a given form geometry. The architectural cost would be representative of any over-specification of materials, finishes and adornments to realize the architectural intentions for the form. Costing information can be obtained directly from the widely available and commercially used Means (1992) Building Construction Cost Data for materials, fabrication, and assembly of the form under consideration, and applied to the design material quantities in question established from the model object data base. For ease of cost data maintenance, and recognizing that relative connector cost fluctuates within a very narrow band with time, object costs can be simply stored as relative values keyed to a cost index. Over time, one only needs to update the cost index to arrive at realistic costs for all objects.

Performance can be measured on the basis of simple demerit rating; the higher the rating from Option criticism, the poorer the Option is in overall performance. The cost of all objects is determined from the sum of the multiples of cost index, relative object cost, and quantity of these objects. Using object relative cost ratios establishes a reasonable time-stable way of storing actual cost information without costly database management for constant updating. Object relative cost ratios can be multiplied by a single cost index value (that can change over time and thus is user-specified) to arrive at the actual cost of objects.

The best joint design Option in terms of cost, or performance, or both, is the Option with the least cost, or performance rating, or both. The ranking of Options is simply done in order of least "distance" as shown in Figure 2.13 to arrive at a list of appropriate design solution suggestions.

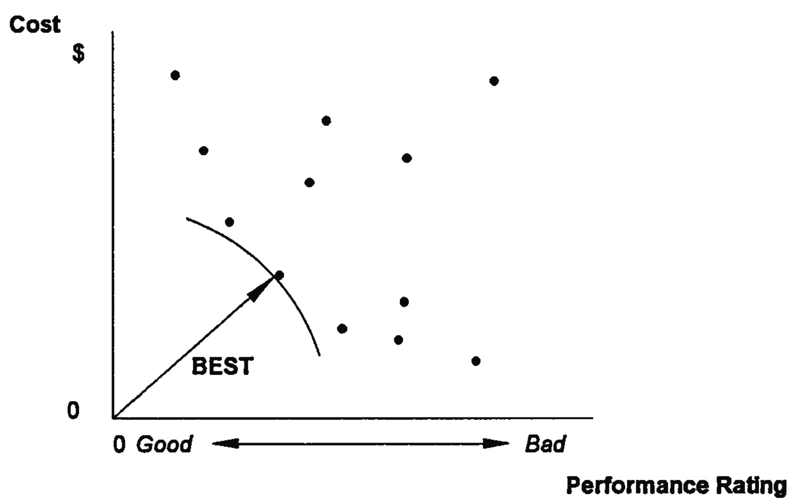


Figure 2.13 Cost/Performance Rating

2.4 USER INTERFACE

Development of a successful user interface infers the simple and elegant handling of numerical, linguistic, and graphical information. If the interface is too complicated, then the user will feel intimidated and will avoid using the design tool. The user interface serves to communicate design requirement information to the problem solver, and should reflect clarity in organization and completeness of information required. Likewise, solution information given back by the system through the interface should be accurate and easily accepted and interpreted.

2.4.1 Object Graphics Modeller

Graphical representation today is so well developed that it remains outside the scope of this thesis for future inclusion. In many instances, graphical information can be distilled into the numerical and linguistic counterparts for analysis, and this is where the current research concentrates. However, the graphics interface is very important and some treatment on its characteristics should be given for future consideration.

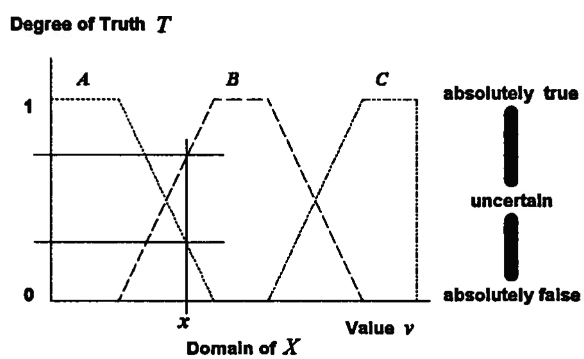
Designers work in a graphical world, more so than any other world. A move to electronic media should very closely replicate that of its real world counterpart to be successful. Ideally, with today's computer object-based graphics technology, it should be possible to manipulate many aspects of a rendered computer simulation of a design artifact as in real life. The system should interface with the user in clear, life-like terms, using simple familiar tools and provide the following functions:

- model objects and their attributes in a 3D context,
- provide easy object description entry, manipulation, 3D viewing, colour / shading / texture functions with corresponding object behavioural response,
- act as the window to the real-time current design state of the constructed model at all times, and
- construct a universal object attribute database that can be easily accessed or modified by external program modules. The current state of the object database should be constantly mirrored to the user through the object modeller at all times.

A number of useful off-the-shelf architectural parametric object modeling packages with the above features currently exist, such as *AutoCAD Designer*™ (Autodesk 1994), that may serve as a suitable modeller provided its parametric object database is easily accessible. Choice of modeller is important as, in addition to the above functions, the modeller interface should very closely replicate the act of exploratory model building by hand using various media, such as cardboard, acetate, plaster, wood sticks, styrofoam, etc. Architects understand that a lot of good information is obtained in the process of model building that is useful in transforming the design to a higher state. A database translator may need to be written to enhance, expand, or clarify the object database. Options of this sort should be considered before any attempt to programming a complete modeller from scratch is undertaken. However, this work attempts to deal with the key parameters of each object so that they can be incorporated into an object graphics modeller.

2.4.2 Communicating with Linguistic Information

Graphical information can often be described linguistically as in the case of visual perception. In order for a qualitative rule system to operate (usually in linguistic terms) with numerical (quantitative) data from a graphic object modeller, some translation between the graphical world (CAD packages) and the linguistic world must take place. This translation activity was found to be an interesting area for research, as such little work in the area was found in the literature (Abella and Kender 1993).



Given x: the outcome is more likely to be B than A.

Figure 2.14 Fuzzy Logic Membership Functions Adaptation

An idea from the field of fuzzy logic was used as the foundation for a numerical/linguistic translator. A membership function shown in Figure 2.14 represents a relationship between validity (formally described as degree of truth) and a range of numerical values. The membership function itself is usually associated with a linguistic label. In essence, the membership function represents a linguistic which has a range of valid numerical equivalents to different degrees.

Membership functions can provide an effective means of dealing with the inflections of meaning in language especially when the meaning is tied to a numerical value. Although most people perceive the same thing in an artifact in general, the degree of the perception is often slightly different between people. This subtle difference yields the richness and variety that designers bring to creative works. Membership functions can model this behaviour quite effectively, as is shown in Chapter 3.

2.5 SYSTEM INTERFACE

In many cases, it is desirable and practical to reserve some activities for specialized procedures done outside the main solution method. Passing information between the main method and the specialized procedure requires an interface between the "systems" that facilitates this activity. The interface can consists essentially of variables or "hooks" that are common to both procedures that can be linked either directly, or through a common data file. System interfaces are useful in linking a development program with one that is currently established, or linking programs of a dissimilar nature, such as analysis programs and CAD programs. A demonstration of this is featured in Chapter 5.

2.6 TEST CASES

Working a manual connection example from start to finish would only prove the difficulty in dealing with the volume and variety of information, and the complexity of all the interrelationships that exist. For this reason, a design example is best left to illustrating the automated form of design proposed by this research. In this manner, concepts covered above can be illustrated to show their effectiveness in dealing with the various aspects of the design problem.

2.7 SUMMARY

Timber connection design falls into the class of configuration problems. Connections are modelled as a kit of parts that are assembled according to a specification language which is partially numerical and partially linguistic in nature. Helpful in this cause is the use of an object-oriented approach that is allegorical to the real world, and the use of linguistic qualitative rules that help formalize the subject matter in terms of simple universal truths. Of important interest here, is the formulation of a plausible design solution for a connection configuration from a set of solution possibilities. This lateral thinking-like process is of benefit to designers, particularly architects, in that a number of interesting design possibilities with criticisms can be considered and presented for a connection in significantly less time and expense that currently only one design could be considered. Two important contributions have come out of the above work, namely: a sharing approach for modular connection representation which is advantageous from a number of viewpoints (see Conclusions), and membership functions for descriptive architectural parameters which can serve as a possible bridge between qualitative and quantitative design.

Chapter 3 QUALITATIVE-QUANTITATIVE DESCRIPTOR TRANSLATION FOR DESIGN

3.1 INTRODUCTION

The areas of architectural and engineering design can be generally classed in nature as qualitative and quantitative, respectively. Typically, quantitative design is numerically based and hierarchically ordered or organized in vertical logical systems; systems that can produce well-defined results from well-defined input if the design problem is well-conditioned. Qualitative design, on the other hand, is normally associative or laterally organized. Because of its illusive nature and poorly conditioned relations between input and output, qualitative design is often framed as the ill-conditioned problem (Oxman and Gero 1987) or even the wicked problem (Bazjanac 1974). Ill-conditioned problems figure prominently in aesthetic design as a designer typically employs a variety of aesthetic moves to create a language to reflect the designer's intended meaning. This form of designing has been interpreted by researchers (Oxman and Gero 1987, Mitchell 1990, and others) as a state-space representation where an initial state is transformed using expert knowledge into a series of solution states. The problem solving is seen as the process of searching through alternative solution states which satisfy certain goals to yield a new design state for the artifact. The recursive interaction of analysis, synthesis, and evaluation continues through the design process.

It is recognized that in order for a purposefully designed artifact to succeed, its qualitative and quantitative roles must be fulfilled. In the case of building design, the roles of architect and engineer have been separated somewhat which has led to misunderstandings in the way each other thinks and speaks in the design sense (Building Arts Forum 1991). However, according to Benjamin (1984), qualitative design thinking and quantitative design thinking historically was not so diverse, and could be moving closer again through the use of the computer. Each mode of thinking can enrich the other, and architects and engineers could profit from an exchange of view points and perhaps even understand each other, if an effort to permit this to happen is advanced. Such a view underlies the topic of this chapter; a bridge between the often linguistically described qualitative and the often numerically described quantitative aspects of design.

Abella and Kender (1993) presented a framework for a system that describes objects in a qualitative fashion using a fuzzification technique to characterize the inherent vagueness of spatial prepositions for describing objects. The research in this thesis is directed towards combining the qualitative and quantitative aspects of design in the framework of an object-oriented environment. The motivation is to represent physical objects as closely to the real-world counterparts as possible, with the addition of intelligence so that objects can interact appropriately in a qualitative and quantitative way consistent with the designer's intentions. Fuzzy logic membership functions offer a promising important bridge between the qualitative (often described linguistically) and quantitative (often described numerically) design worlds. The intentions for the use of fuzzy logic membership functions in qualitative design (aesthetics) are presented herein, and experimental evidence that follows appears to support the validity of this idea. The smart object idea has been around for a while, but not in connection with the use of fuzzy logic and dynamic membership functions. In particular, this thesis proposes that fuzzy logic is an appropriate technology for dealing with the ill-conditioned design problem of aesthetics. Some background on the technology and its application to aesthetics is briefly discussed. Two detailed applications and study results using the proximity of objects and the interpretation of colour are presented in order to demonstrate the viability of the concept.

3.2 BACKGROUND ON FUZZY LOGIC MEMBERSHIP FUNCTIONS

A good in-depth coverage of fuzzy logic and fuzzy set theory is available in textbooks such as Zimmerman (1991) and Kosko (1992), and seminars (Motorola Fuzzy Logic 1993), therefore only relevant concepts will be briefly summarized here. Fuzzy logic was conceived in 1965 by Professor Lotfi Zadeh at UC Berkeley to help the inexact area of socio-economic research. It is an extension of classical set theory and is supported by rigorous mathematics. The term *fuzzy* equates to imprecise, like natural language adjectives. Fuzzy logic is simple, intuitive, and robust to system non-linearity, but is unsuited for precise calculations.

Applications of fuzzy logic technology seem to be commonplace today in such examples as anti-lock brakes, camera-lens focusers, HVAC controls, etc. Fuzzy logic is particularly suited to device controls where the relationship between input and output is poorly understood or contextually variant. The technology has proven to be effective in dealing with these kinds of issues and is well presented in the literature (Motorola Fuzzy Logic 1993, Sibigtroth 1992, Zimmerman 1991).

Central to the fuzzy logic model is the membership function. Membership functions represent the degree of membership a numerical value has throughout its "membership" domain. As shown in Figure 3.1, membership

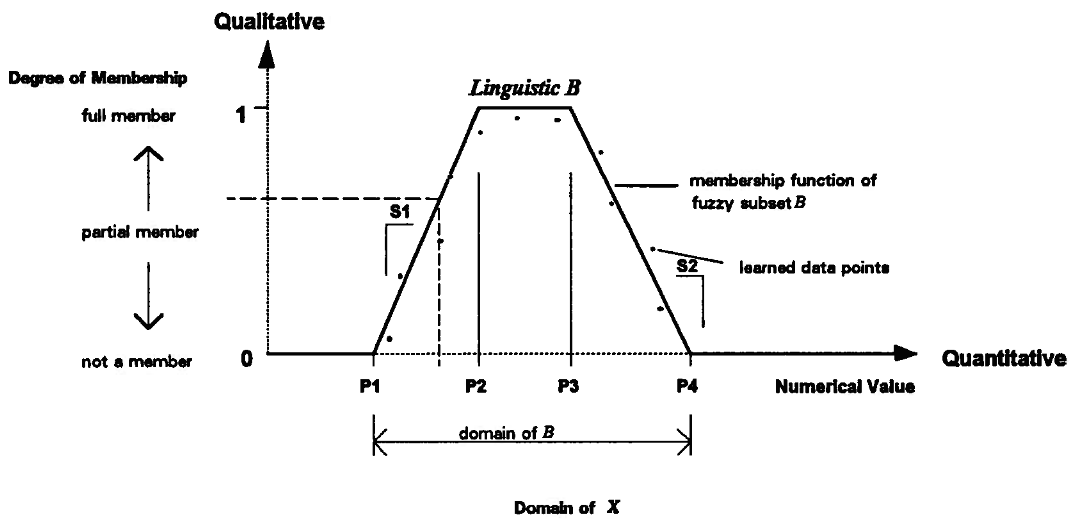
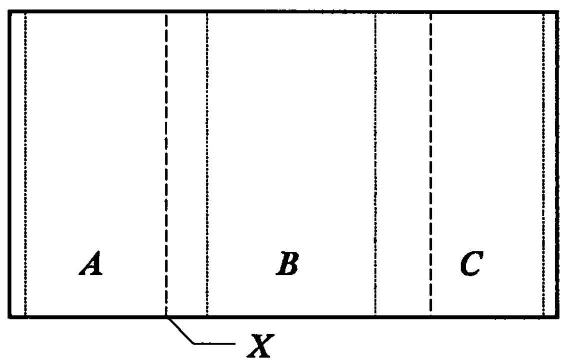


Figure 3.1 Simple Trapezoidal Membership Function Anatomy

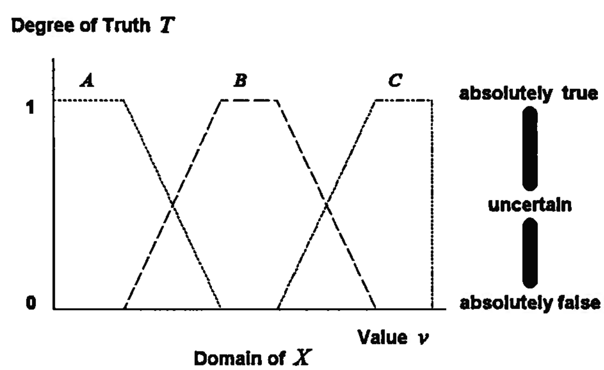
functions can be simply described as triangular, trapezoidal, Gaussian, etc., or if discrete data is used, a curve-fit. They are added to the domain in relation to additional possible "membership" descriptors, each covering a particular domain range. Membership functions become an effective and important translator between numerical and linguistic information. There are several significant properties here which will be exploited later in the context of the problem at hand.

3.3 THEORY OF APPROXIMATION FOR DESCRIPTIVE VARIABLES IN A DESIGN SPACE

Consider three descriptive variables A, B, C within a design space X, as shown in Figure 3.2. Each variable is typically at times vague in its meaning ($A \cap B$, $B \cap C$), while at other times absolute in its meaning (A, B, or C). The meaning of each variable can be attached to a measurable quantity v. In this design space, descriptive variables may only take on two possible meanings in their scope of vagueness, that is, the design space descriptors transcend from one variable, to another, in succession.



Plan View



Elevation View

Figure 3.2 Representation of Descriptive Variables within a Design Space

Using a concept from fuzzy set theory, each descriptive variable can be represented as a membership function of degree of truth T against a measurable quantity v. The domain of the membership function is the possible measurable quantities within the design space X. Hence for n descriptive variables, there would be n corresponding membership functions covering the domain X.

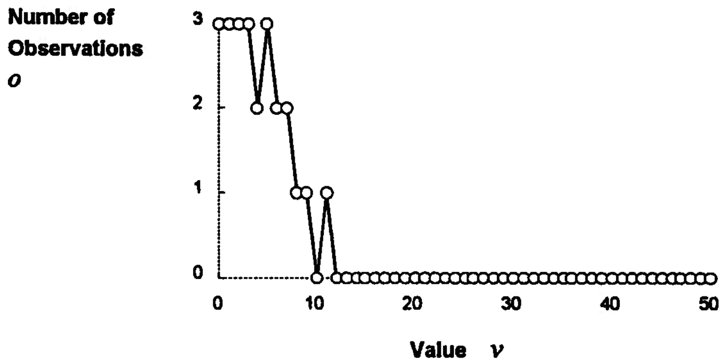
Each descriptive variable has measurable quantities for which the descriptive meaning is absolutely true (1), absolutely false (0), or uncertain ($0<T<1$). However, the truth divisions in themselves may be uncertain, and therefore can be known with the same degree of normal probability as case data is accumulated.

3.4 CONSTRUCTION OF QUALITATIVE MEMBERSHIP FUNCTIONS

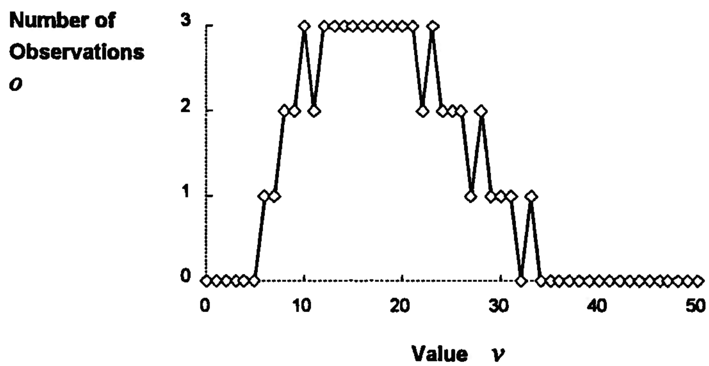
A fuzzy logic membership function describes a relation between the degree of membership (truth) T to a classification and a numerical parameter v in the context of a linguistic classification. From tests where subjects are presented with a ranging variety of cases (for which numerical quantities v are known) to classify in succession, it is possible to ascertain basic membership function properties of each classification. Each case in a test is prepared aforehand to correspond directly to a numerical whole number value v. Fifty cases covering the domain were deemed sufficient to give a reasonable representation of the domain and also provided a means to reduce the absolute domain value v to a dimensionless quantity in the range of whole numbers from 0 to 50. Hence, each linguistic membership function can have discrete values of T for each v from 0 to 50. In a test, a subject classifies each suite of 50 randomly presented cases into corresponding linguistic classifications. In the process, numbers of occurrences o accumulate for each value v for each linguistic membership function according to subject classification. Figure 3.3 shows a set of plotted classified data after 3 test suites. With repeated test suites, the number of classed occurrences o increases for each value v for each linguistic classification, representative of learning by experience. At any point, the degree of truth T to a classification for a given value v for a particular membership function M_i is reduced from the raw classification occurrences data o by:

[3.1] $$T_i(v) = \frac{o_i(v)}{\sum_{j=1}^{n} o_j(v)}$$

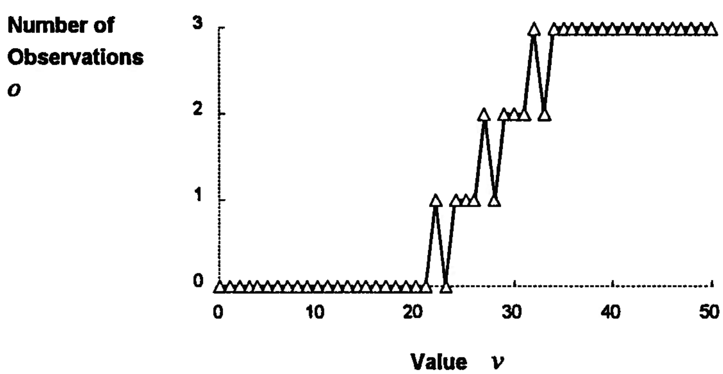
where n is the total number of linguistic membership functions in the domain.



Classification A



Classification B



Classification C

Figure 3.3 Classified Data Point Accumulation

From the reduced test data, key membership function parameters *P1, P2, P3, P4, S1, S2* in Figure 3.1 are obtained. The regions of the data where 0<*T*<1 are of special interest, since these regions represent uncertainty in classification; a perhaps subjective characteristic of decision-making. By looking at the regions of doubt in Figure 3.4 where successive classifications have truth values other than 0 or 1, the region of doubt can be modeled by a simple crossing of the member functions, with the intersection point at a truth value of 0.5. This process is repeated for each succeeding pair of classifications to yield a set of classified membership functions for the subject description of interest.

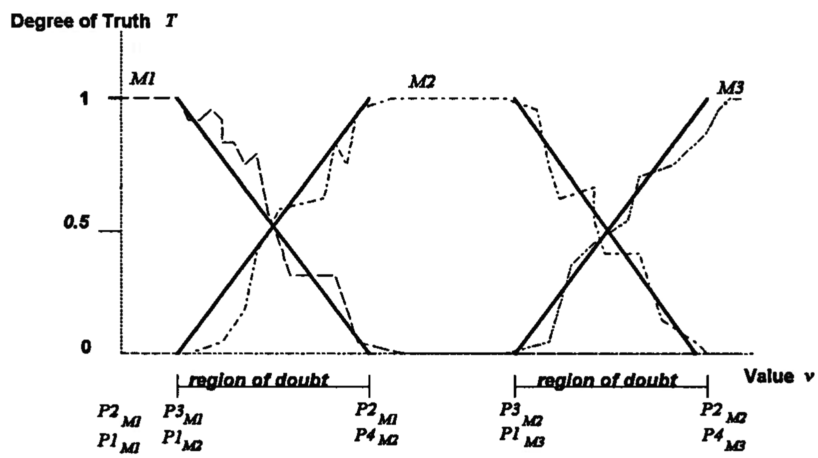


Figure 3.4 Membership Function Construction from Doubt Regions

The region of doubt between two adjacent membership functions is easily found as the set of values *v* of both membership functions for which corresponding *T* values in the range 0<*T*<1 exist. The membership function intersection at a truth value of 0.5 is fixed at the mean of the values *v* within each region of doubt, and the *P1, P2, P3* and *P4* values of the membership functions at truth values of 0 and 1 are placed at the corresponding 5th and 95th percentiles respectively of the normal distribution of all values *v* within each region of doubt using:

[3.2] $$P1 = P3 = \bar{x} - 1.645\sigma$$

and

[3.3] $$P2 = P4 = \bar{x} + 1.645\sigma$$

where x is the list of values v within the doubt region, n is the number of values in x, $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$ is the mean of x, and $\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$ is the standard deviation of x.

The corresponding slope values are found from:

[3.4] $$S1 = \frac{1}{P2 - P1}$$

and

[3.5] $$S2 = \frac{1}{P3 - P4}.$$

Such a membership function construction method proved simple and effective in the studies which follow since no standard methods were found in the literature. In addition, raw classified occurrence data can be added at any point at any time, and when reduced by the above method, updates the current set of membership functions. This ability to update at anytime with new input data is what is meant by *dynamic membership functions* here; a process allegorical to learning by experience. The data may be new, but its use is buffered by the experience of the old.

3.5 USAGE OF MEMBERSHIP FUNCTIONS IN QUALITATIVE DESIGN

Qualitative design employs primarily a visual media form, however it is often critiqued with linguistics. Words can capture the essence of visual experiences using descriptors that most people have a definition of on a more or less common basis. The subtleties in descriptive values among beholders and designers give a certain richness and identity to the artifact, and to the subject of aesthetic design. It is important that these subtleties be captured somehow so that a designer's individual creativity is preserved and not dictated or constrained by prescription. Hence, a number of foreseeable stages to the development of an aesthetics facility using fuzzy logic are apparent:

1. The translation between qualitative (linguistic) and quantitative (numerical) ill-conditioned information.
2. The manipulation of aesthetic design information in linguistic terms to form a visual language for design.

3. The manipulation of language information, possibly referencing Gestalt rules, to form meanings of designed artifacts.

The first point, which is probably the most important for establishing the bridge between qualitative and quantitative information, is dealt with in this thesis. The basics of aesthetics in design in general are well presented by Edel (1967), and by Holgate (1992) and Isaac (1971) for architectural design in particular.

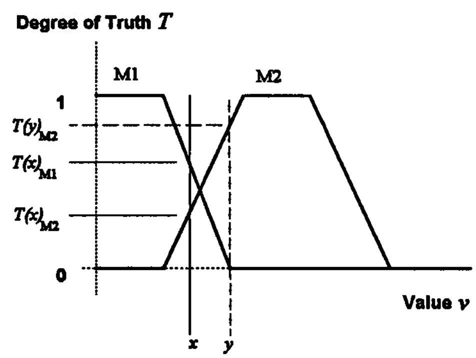
In the context of the translation problem, two features of membership functions are useful. Firstly, one typically learns a task by experience and acquires a certain competency along the way. This process tends to build confidence in knowing what task words or descriptors actually mean. Membership functions can be described using a set of points and subsequent curve-fit; points which signify a certain event that has been "classed" by the user. Over time, the user can classify a number of numerical values for this membership function, thus strengthening its descriptive meaning. This process closely replicates learning by experience. The idea of a membership function with a growing data base can be referred to, as mentioned previously, a *dynamic membership function* because of its ability to alter its description/meaning as experience is gained. Eventually, the overall shape of the acquired data will take on the familiar linear tent-shape as previously described. This meaning of dynamics is different from the one expressed by Smith and Takagi (1993) for dynamic reasoning method switching.

The second feature of note is that it is possible to detect a level of certainty to which an object with a certain value will belong to a particular membership class. If the membership function is described as a noun, then this step is providing an adjective. Figure 3.4 shows how. Consider two membership functions M1 and M2, and their truth values $T(v)_{M1}$ and $T(v)_{M2}$, for a given input value x. By weighing their truth values $T(x)_{Mi}$ in Bayesian terms as:

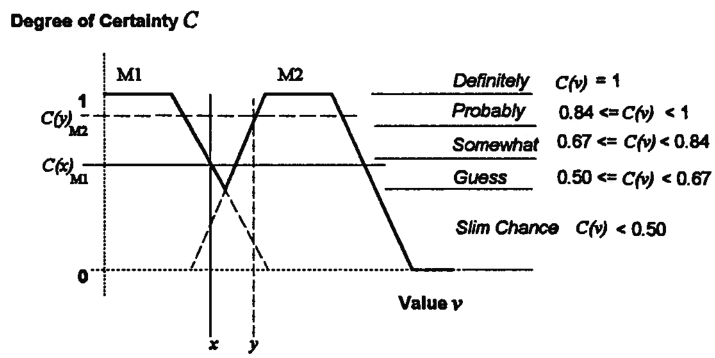
[3.4]
$$C(x)_{Mi} = \frac{T(x)_{M_i}}{\sum_{j=1}^{n} T(x)_{M_j}}$$

where n is the number of member functions in the domain, it is possible to assign a qualitative level of certainty from $C(x)$ for a truth value which would be a reflection of *how true* the input value fits a particular membership class. This feature adds another linguistic dimension to the subtlety of classification in that the certainty level assists in ranging a quantitative value within a qualitative meaning. Usually, the largest truth value, and therefore the degree of certainty of this truth, is of interest as a linguistic indicator of ***how well*** the value relates to the corresponding membership linguistic.

As a quantitative to qualitative example, for a quantitative value x in Figure 3.5, the corresponding



Membership Functions



Certainty Functions

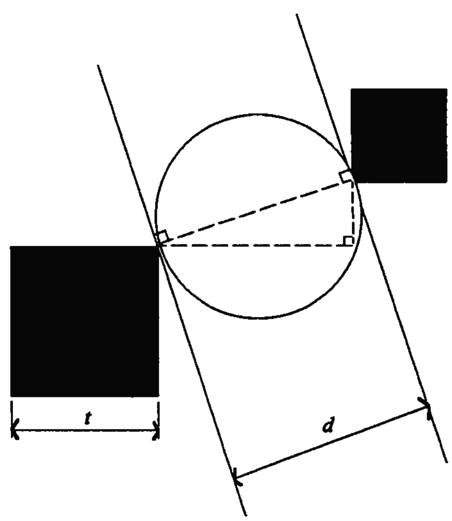
Figure 3.5 Certainty Functions

qualitative description would be *Guess M1*. Alternately, a qualitative description of *Probably M2* would produce a quantitative value of y and a corresponding physical result. Note that this value of y came from the left sloping limb of *M2*, which raises an interesting point. The value y could just have easily come from the right sloping limb of *M2* resulting in an entirely different but equally valid value for y. The choice of membership function limb correlates to the context for the desired result which must be known *apriori* in order for the correct value of y to be suggested by the membership function. In other words, if two values of y can result from a linguistic meaning, context must be used in order to select the correct value. Currently, context is simply handled by presenting two physical results for corresponding values of y for external selection.

3.6 PROXIMITY STUDY MEMBERSHIP FUNCTIONS

The objective of the proximity study was to essentially establish the fuzzy logic membership function adaptation as a suitable translation model between ill-conditioned numerical and linguistic data.

The concept of proximity has applications in the aesthetic grouping of objects (fasteners or structural members). A software application, PROXIMIZER, was developed to capture membership functions based on a few linguistic classifications of proximity as interpreted by the user, and use these relations to evaluate the proximity of two boxes in linguistic terms. Drawing from Isaac (1971) and Prak (1968), the definition of proximity used related the minimum distance between two bodies to the smallest dimension of the largest body, as shown in Figure 3.6. This



$$proximity\ ratio = \frac{d}{t}$$

Figure 3.6 Proximity Ratio Nomenclature dimensionless ratio was placed on a scale from 0 representing touching boxes, to 50 representing extremely separated boxes. Figure 3.7 shows a screen capture of PROXIMIZER after assessing the proximity of a pair of same-sized boxes based on the current set of data used to determine the linguistically-labelled membership functions. Should the user choose to disagree with PROXIMIZER's linguistic response, the user has the choice to either add an additional set of calibration test data to refine the complete current membership function database, or the user can dynamically add the single proximity value point in dispute to the user's choice of available linguistic membership function classifications. In this manner, the user can fine tune the membership function data base to a degree that a consistent fit between observed proximity cases and linguistic evaluations results.

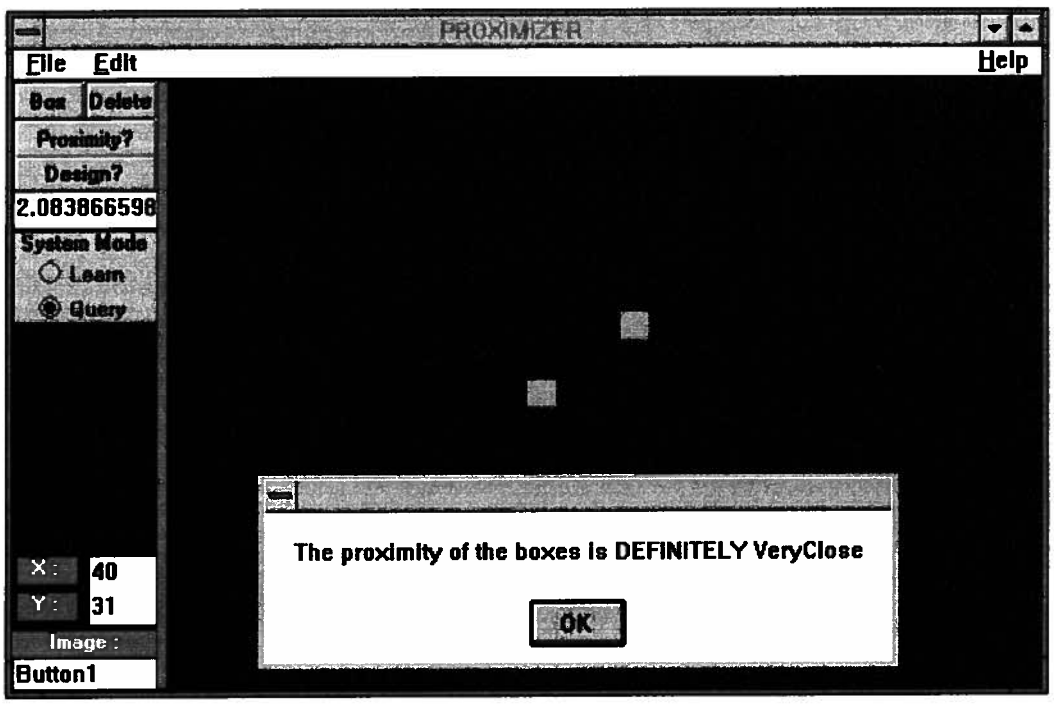
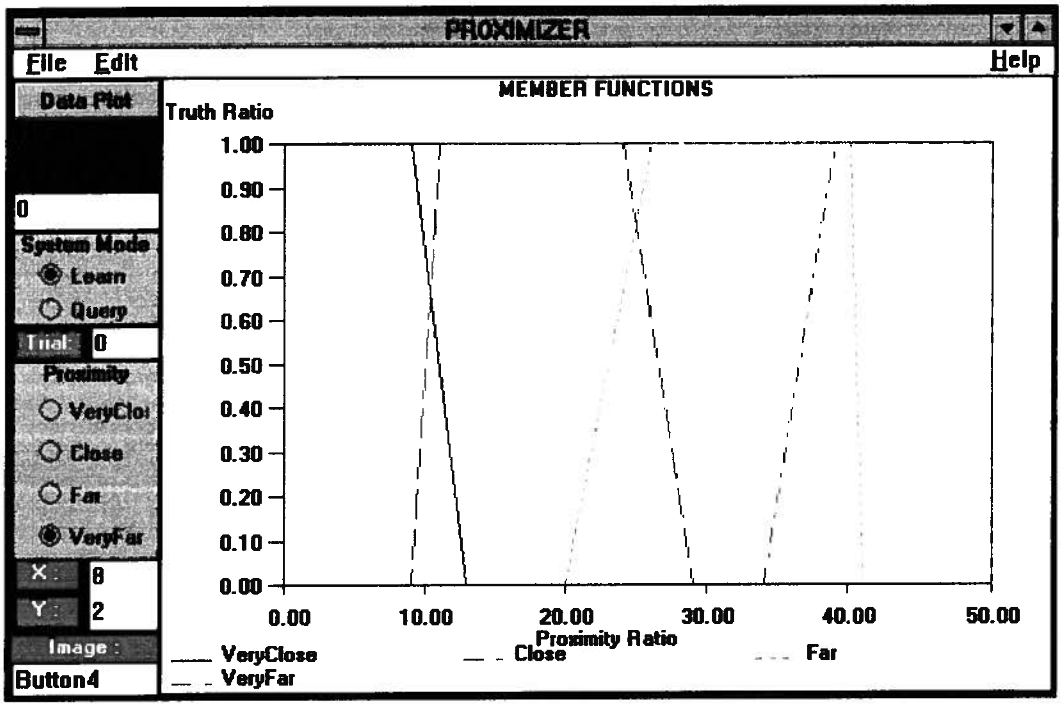


Figure 3.7 PROXIMIZER Determination of the Proximity of Two Boxes

PROXIMIZER is also capable of design by arranging two boxes in positions prescribed by a linguistic command. The knowledge base is dynamic in the sense the knowledge base can grow as desired based on proximity case data accumulated (or added as shown above), similar to learning by experience. Membership functions, shown in Figure 3.8, are used in thetranslation between linguistic (qualitative)and measurable (quantitative) data.



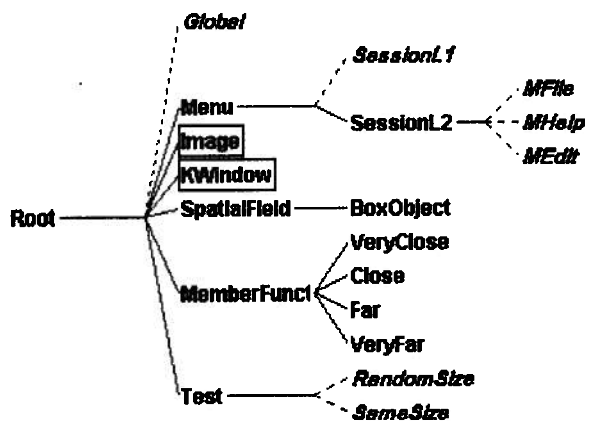


Figure 3.8 PROXIMIZER Membership Function Display and Object Structure

The membership functions should be similar but not exactly the same between two people since individuals rarely see things exactly the same way. If the membership functions are similar, then a correct measuring parameter had been chosen as the quantitative variable. The membership functions used in PROXIMIZER are the basic linear tent-shape models as shown before. In further work, other membership function shapes may also be appropriate and updated according to learned data. This dynamic approach has the benefit of consolidating a design term (and thus its subjective meaning) with additional data, much in the same way a person learns by experience. PROXIMIZER was developed further so that given linguistic input, a corresponding visual design case output could be produced (design mode); or given a visual design case, a corresponding linguistic output could be ascertained (analysis mode). The linguistic input/output not only contained the classification, but also its certainty description.

To assess the above approach, a membership function study using PROXIMIZER was undertaken. A test group of 20 engineering students (2 males with arts/architecture background, and 15 males and 3 females with science/engineering background) performed concurrently a set of explicit operations with PROXIMIZER from a detailed instruction sheet. The test subjects used PROXIMIZER to subjectively grade two sets of box proximities: one set consisting of two square boxes the same size randomly located in a spatial field, and another set of two boxes of random size, shape, and spatial field location. Fifty box proximity cases comprised each set. On completion of the test sets, PROXIMIZER constructed membership/certainty functions using the classed data from each subject. The students took about 40 minutes to complete the instruction set.

Data files mentioned in the instruction sheet were gathered from the computers for analysis. Each data file captured key membership function points, and raw and relative truth data with respect to proximity ratios from 0 to 50 for the classifications of *VeryClose*, *Close*, *Far*, and *VeryFar*, respectively. The membership functions from all test subjects for each grading were then compared to assess their agreement. If general agreement was found, then the quantitative measuring parameter had been chosen correctly implying that the subject designers were in general agreement with the definition and subtleties of proximity concepts. If poor agreement existed, then perhaps the measuring parameter was at fault and needed adjusting, or test subjects were not assessing the cases accurately.

In analyzing the data obtained, one could explore all kinds of effects, such as gender, design background and preference, and classification repetition. However, in the interest of time and the consideration of the suitability goal, it is useful to deal in generalities and thus reduce the data to simple aggregate averages with no weighting. The data revealed the results in Figure 3.9 plotted from the average of all student classified responses

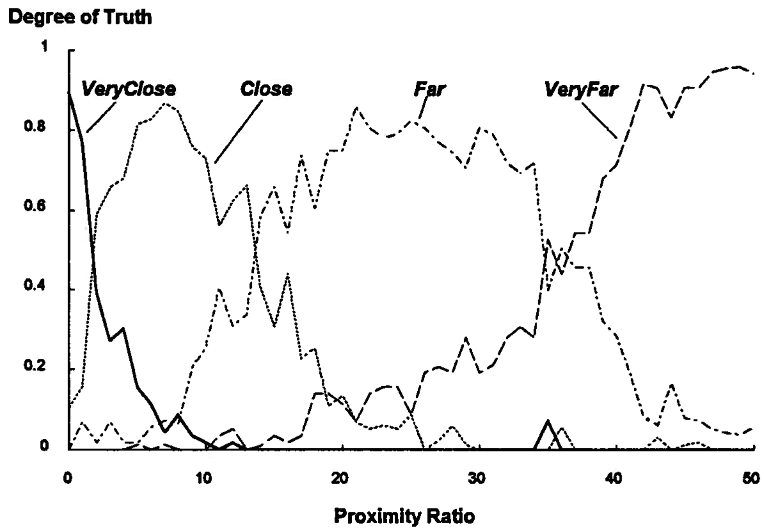


Figure 3.9 Average Classified Proximity Data - Same Box Size, Aspect Ratio

from the third PROXIMIZER cycle for same size boxes (SS3). The scatter of all classified data was visually represented quite well by the averaged relations given above. Of immediate note is that the data is strongly segregated into the four classifications, with little overlap in the odd-sequenced classes (*Close* and *VeryFar*). Thus it would appear that the quantitative measuring model is generally correct. The odd-sequenced overlap can be a result of classification judgment error on the part of some students, however it is quite small. The maximums of the peaked classification curves are close to a degree of truth of 1 (at least over 0.9) indicating that the quantitative-qualitative model is indeed believable. The shape of the test data distribution can be simplified into a linear tent-shape membership function form for each data classification as in Figure 3.10 using the method as outlined previously.

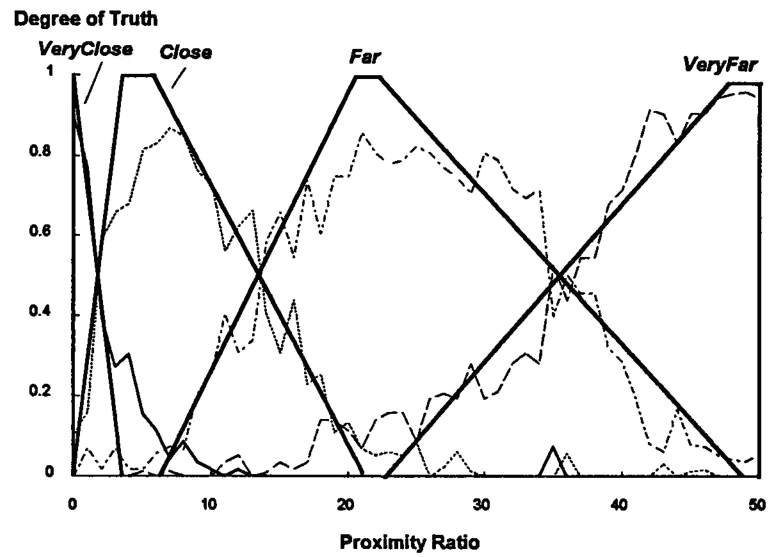


Figure 3.10 Membership Function Fit to Data - Same Box Size, Aspect Ratio

The model can become dynamic in the sense that if more data is added, the membership function points *P1, P2, P3, P4, S1, S2* in Figure 3.1 can shift to reflect this acquisition of new knowledge. Over time, the shifts would become progressively minor, reflecting the "experience" influence.

The averaged PROXIMIZER data in Figure 3.11 for the third series (RS3) random size boxes illustrate

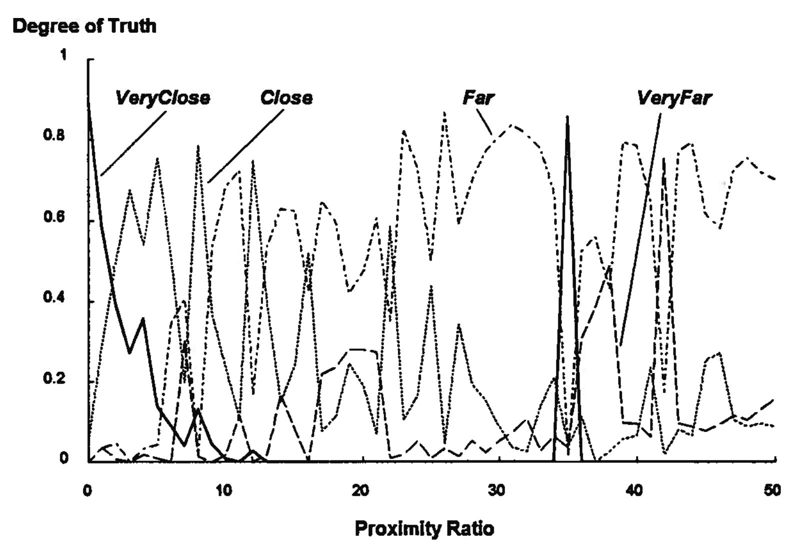


Figure 3.11 Average Classified Proximity Data - Random Box Size, Aspect Ratio

another point. The classifications here appear to be less segregated, with lower-valued peaks for degree of truth, possibly reflecting general confusion in the judgment of the test subjects. The confusion may center in the fact that the addition of changing shape and changing aspect ratio, both separately measurable qualitative variables in their own right, were not properly accounted for in the quantitative measuring model. Rather, the characteristics of size, and aspect ratio, should have separate classified membership functions associated with them which, when combined collectively as in Figure 3.12 with the previous same size proximity model, could yield a correct fuzzy

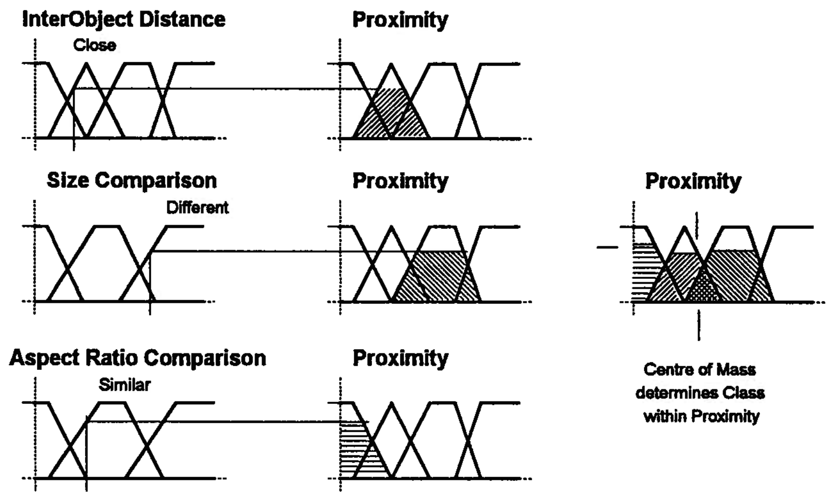
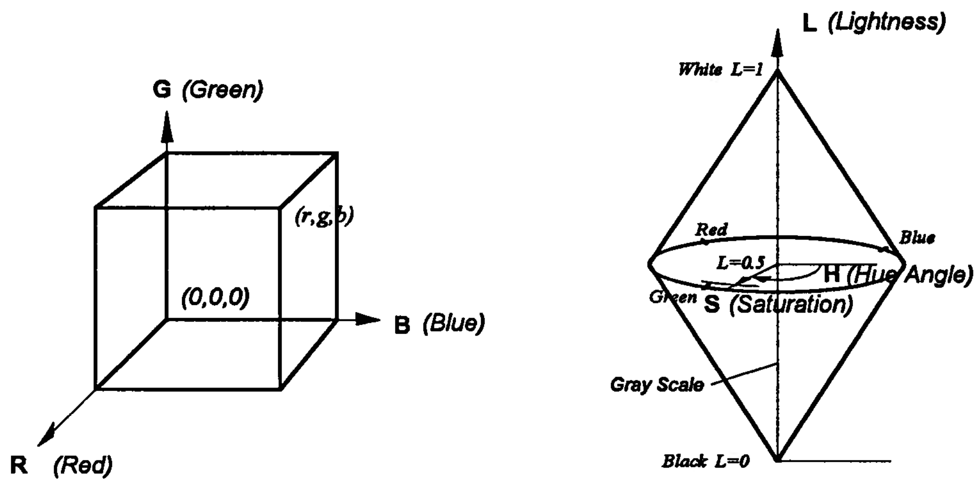


Figure 3.12 Fuzzy Inference Through Membership Functions

logic model for the proximity of variably sized and shaped boxes. This remains to be experimentally shown and is probably the logical next step for verification. However, the important issue of establishing a suitable qualitative-quantitative model template for ill-conditioned design problems appears close to resolution in the terms of the initial outlined objective.

3.7 COLOUR STUDY MEMBERSHIP FUNCTIONS

The objective of the colour study was to further establish the fuzzy logic member function adaptation as a suitable translation model between ill-conditioned numerical and linguistic data. The concept of colour is universal in its definition yet can have subtle variations in linguistic description that can lead to richness in design. A software application, COLOURIZER, was developed to capture membership functions based on a few linguistic classifications of colour as interpreted by the user, and use these relations to evaluate the colour of a coloured box in linguistic terms. Drawing from Glassner(1990), Hall(1988), and Hearn and Baker (1986), the definition of colour used related the red-green-blue



RGB Colour Cube *HSL Double Cone*

Figure 3.13 Colour Representation Models

(RGB) values of a colour viewed on a computer monitor screen to a commonly accepted linguistic description of colour defined in terms of *Hue*, *Saturation*, and *Intensity (Lightness)* as in Figure 3.13. The RGB colour space is used commonly in electronic video equipment and is intregal to its colour-producing electronic components, while the HSL colour space is an accepted familiar norm for describing colour among artists. *Hue* characterizes the base identity of the colour as it exists in the natural visible light spectrum, and is commonly represented as the Munsell colour wheel in fine arts literature (see Isaac 1971). *Saturation* characterizes the purity of the colour in a scale ranging from muddy gray to pure colour. *Intensity* or Lightness characterizes the luminescence (shade) of the colour on a scale ranging from black to white.

Since the colour descriptions (membership functions) were to be based on linguistic terms from the HSL colour space, quantitative translation between the computer-represented RGB colour space and the more familiar

HSL colour space was required. Figure 3.14 shows the quantitative translation functions from RGB to HSL representation and vice-versa, as derived from Glassner (1990), that were used. All qualitative-quantitative relations were constructed in the HSL colour space.

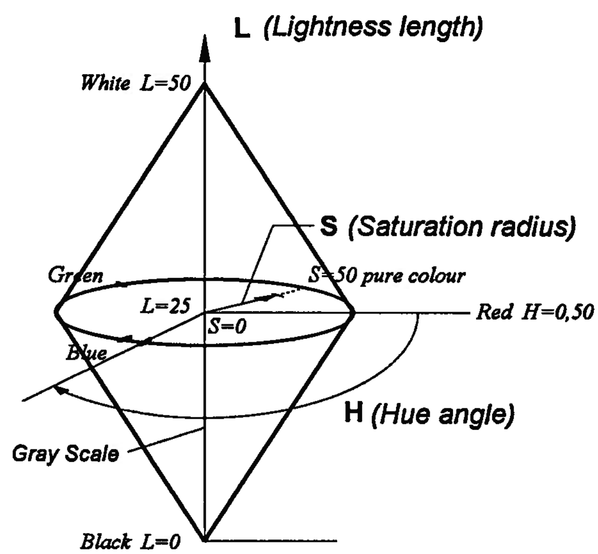
```
Method RGBtoHSL
{
Box:R = GetNthElem(Bitmap1:BackgroundColor,1);
Box:G = GetNthElem(Bitmap1:BackgroundColor,2);
Box:B = GetNthElem(Bitmap1:BackgroundColor,3);
ConvertRGBtoHSL(Box,Box:R/255,Box:G/255,Box:B/255,240);
};

Function ConvertRGBtoHSL (item r g b range)
{
item:ValueIntensity = (Max(r,g,b) + Min(r,g,b)) / 2.0;

If (item:ValueIntensity != 0)
Then (item:ValueSaturation = Max(r,g,b) - Min(r,g,b))
Else item:ValueSaturation = 0;

If (item:ValueSaturation != 0)
Then {
      If (item:ValueIntensity <= 0.5)
      Then (item:ValueSaturation = item:ValueSaturation / (Max(r,g,b) + Min(r,g,b)))
      Else item:ValueSaturation = item:ValueSaturation / ( 2.0 - Max(r,g,b) - Min(r,g,b));
      };

If (Max(r,g,b) == r)
Then {
      If (Min(r,g,b) == g)
      Then (item:ValueHue = 5 + (Max(r,g,b) - b) / (Max(r,g,b) - Min(r,g,b)))
      Else item:ValueHue = 1 - (Max(r,g,b) - g) / (Max(r,g,b) - Min(r,g,b));
      }
Else If (Max(r,g,b) == g)
      Then {
            If (Min(r,g,b) == b)
            Then (item:ValueHue = 1 + (Max(r,g,b) - r) / (Max(r,g,b) - Min(r,g,b)))
            Else item:ValueHue = 3 - (Max(r,g,b) - b) / (Max(r,g,b) - Min(r,g,b));
            }
Else If (Min(r,g,b) == r)
      Then (item:ValueHue = 3 + (Max(r,g,b) - g) / (Max(r,g,b) - Min(r,g,b)))
      Else item:ValueHue = 5 - (Max(r,g,b) - r) / (Max(r,g,b) - Min(r,g,b));

item:ValueHue = item:ValueHue / 6 * range;
item:ValueSaturation = item:ValueSaturation * range;
item:ValueIntensity = item:ValueIntensity * range;
};
```

Figure 3.14 COLOURIZER RGB-HSL Quantitative Translation Functions

```
Method HSLtoRGB
{
ConvertHSLtoRGB( Box, Box:ValueHue/240, Box:ValueSaturation/240, Box:ValueIntensity/240, 255 );
ClearList( Bitmap1:BackgroundColor );
AppendToList( Bitmap1:BackgroundColor, Box:R, Box:G, Box:B );
ResetImage( Bitmap1 );
};

Function ConvertHSLtoRGB (item h s l range)
{
Global:v = (If ( l <= 0.5) Then (l * (1.0 + s)) Else l + s - l * s );

If (Global:v <= 0 )
Then {
        item:R = 0;
        item:G = 0;
        item:B = 0;
        }
 Else {
        If (Floor( 6 * h ) == 0)
        Then {
        item:R = Global:v;
        item:G = (l + l - Global:v) + 2 * (Global:v - l) * (6 * h - Floor( 6 * h ));
        item:B = l + l - Global:v;
        }
        Else If (Floor( 6 * h ) == 1)
        Then {
                item:R = Global:v - 2 * (Global:v - l) * (6 * h - Floor(6 * h));
                item:G = Global:v;
                item:B = l + l - Global:v;
                }
        Else If (Floor(6 * h) == 2)
        Then {
                item:R = l + l - Global:v;
                item:G = Global:v;
                item:B = (l + l - Global:v) + 2 * (Global:v - l) * (6 * h - Floor(6 * h));
                }
        Else If (Floor(6 * h) == 3)
        Then {
                item:R = l + l - Global:v;
                item:G = Global:v - 2 * (Global:v  - l) * (6 * h - Floor(6 * h));
                item:B = Global:v;
                }
        Else If (Floor(6 * h) == 4)
        Then {
                item:R = (l + l - Global:v)+ 2 * (Global:v - l) * (6 * h - Floor(6 * h));
                item:G = l + l - Global:v;
                item:B = Global:v;
                }
        Else If (Floor(6 * h)  == 5)
        Then {
                item:R = Global:v;
                item:G = l + l - Global:v;
                item:B = Global:v - 2 * (Global:v  - l) * (6 * h - Floor(6 * h));
                };
        };

    item:R = item:R * range;
    item:G = item:G * range;
    item:B = item:B * range;
    };
```

Figure 3.14 *continued*

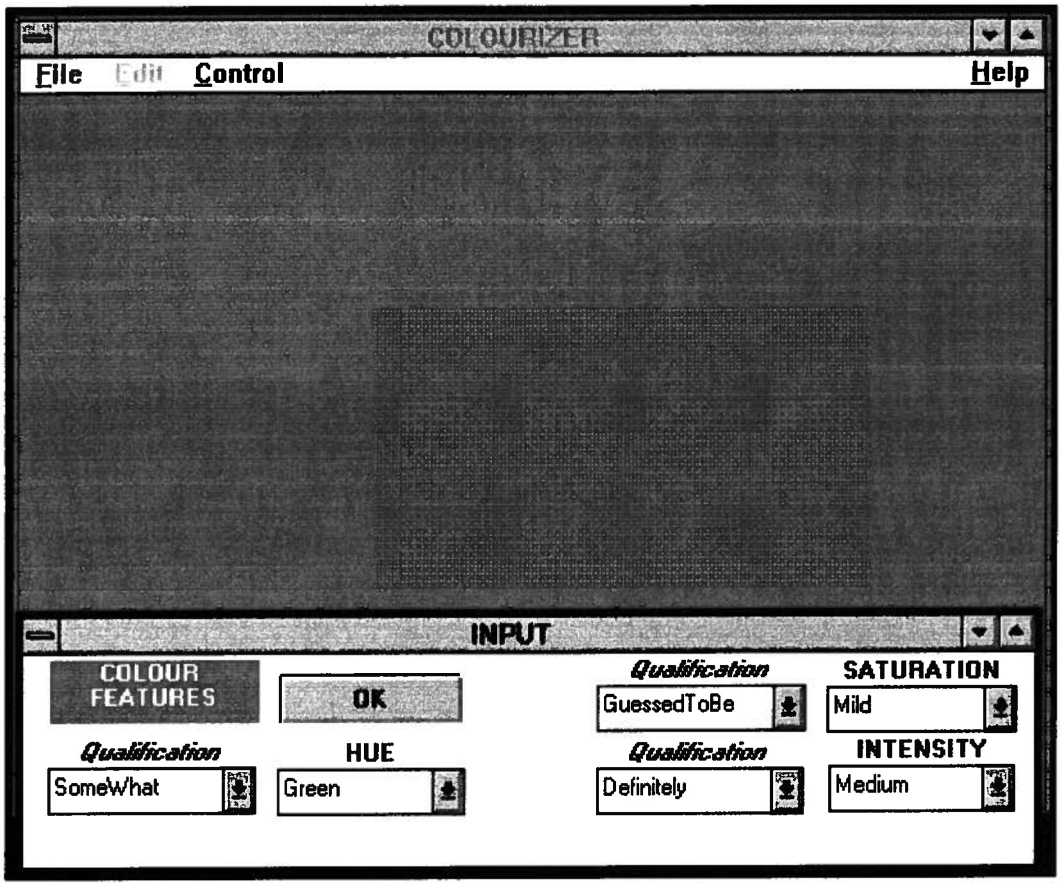
The quantitative values of *Hue*, *Saturation*, and *Intensity* were then scaled to ranges between 0 and 50 respectively as shown in Figure 3.15.



HSL Double Cone

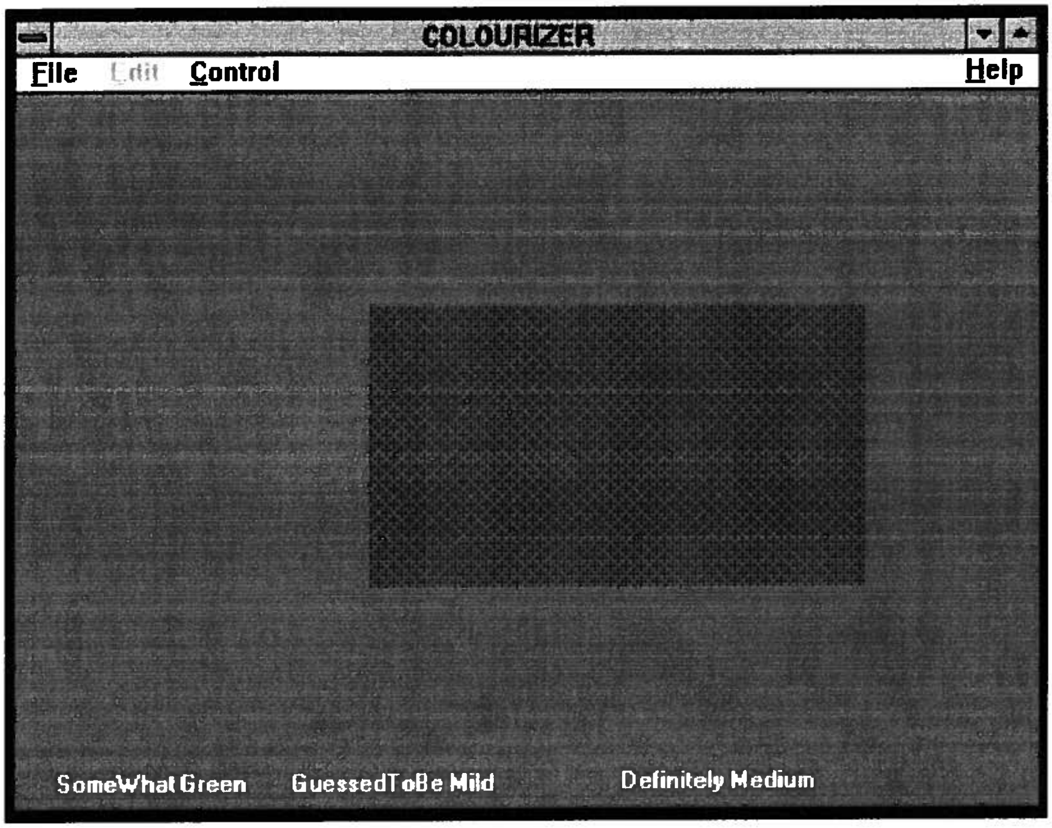
Figure 3.15 COLOURIZER HSL Quantitative Description

COLOURIZER, with similar functional features to PROXIMIZER, was also capable of design by showing colours as prescribed by a linguistic command as in Figure 3.16, or if given a colour, COLOURIZER can describe it in words. Dynamic refinement of the membership function data base as featured in PROXIMIZER allows the user the ability to shape the membership functions to a personal interpretation. In this sense, the knowledge base is dynamic in that it can grow as desired based on colour case data



create colour as defined by user

Figure 3.16 COLOURIZER Linguistic Determination of Colour Attributes



linguistically describe colour created by user

Figure 3.16 *continued*

accumulated, similar to learning by experience. Membership functions, shown in Figure 3.17, are used in the translation between linguistic (qualitative)and measurable (quantitative) data. The membership functions should be similar but not exactly the same between two people since individuals rarely see things exactly the same way. If the membership functions are similar, then a correct measuring parameter had been chosen as the quantitative variable. The membership functions used in COLOURIZER are the basic linear tent-shape models constructed as shown before. This dynamic approach has the benefit of consolidating a design term (and thus its subjective meaning) with additional data, much in the same way a person learns by experience. COLOURIZER was developed further so that given linguistic input, a corresponding visual design case output could be produced (design mode); or given a visual design case, a corresponding linguistic output could be ascertained (analysis mode). Again, the linguistic input/output not only contained the various colour aspect classifications, but also their corresponding certainty descriptions.

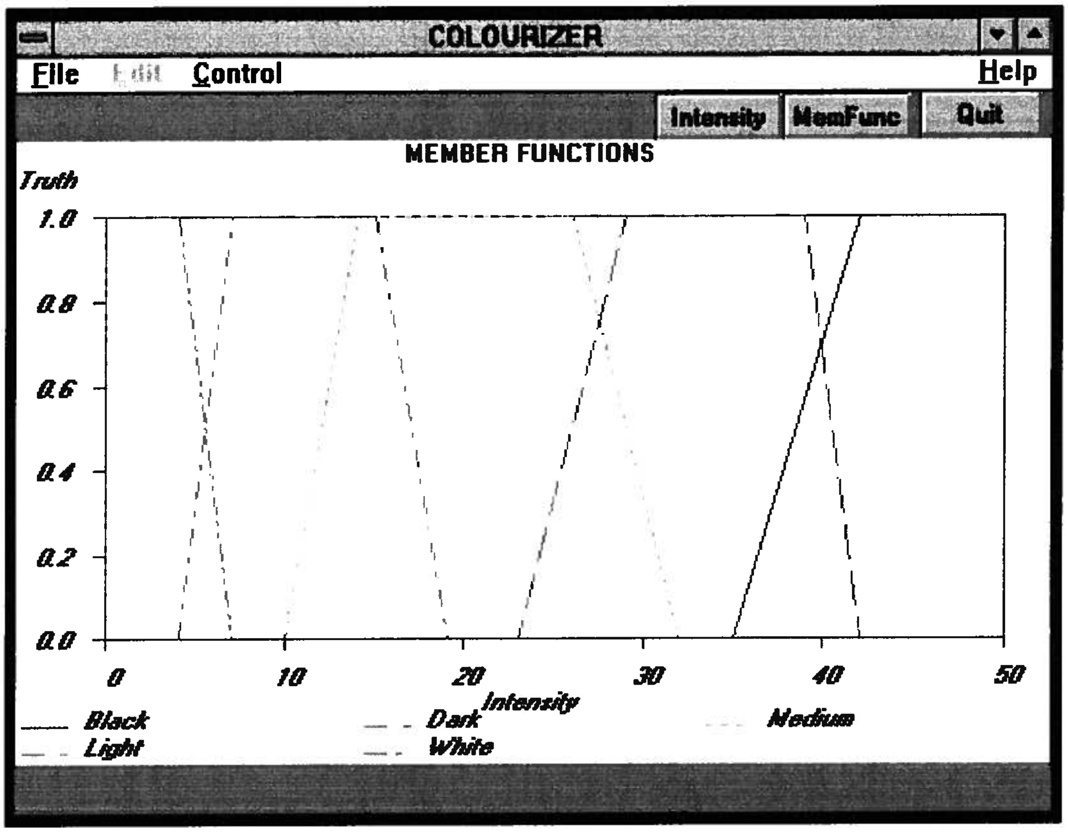


Figure 3.17 COLOURIZER Membership Function Display and Object Structure

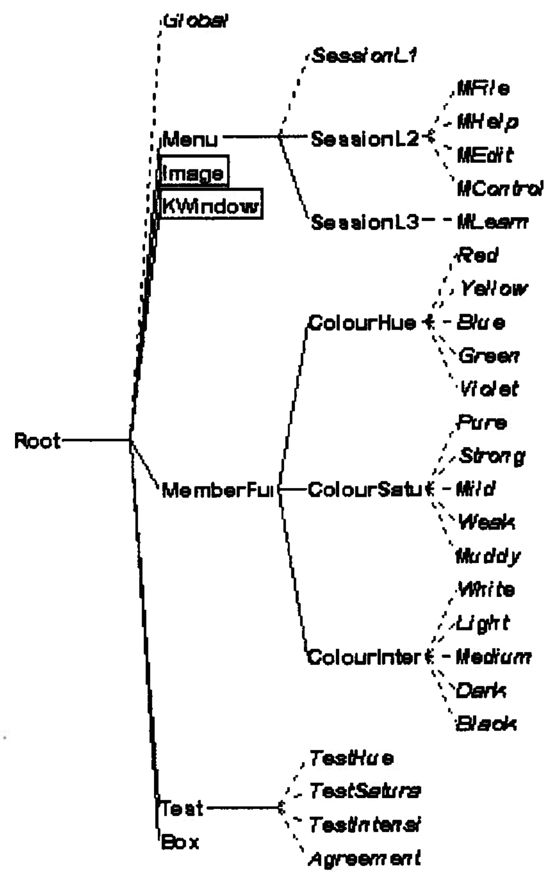


Figure 3.17 *continued*

To assess the above approach, a membership function study using COLOURIZER was undertaken. A test group of 30 engineering students (1 male and 1 female with arts/architecture backgrounds, and 22 males and 6 females(1 colour blind) with science/engineering backgrounds) performed concurrently a set of explicit operations with COLOURIZER from a detailed instruction sheet. The test subjects used COLOURIZER to subjectively grade the hue, saturation, and intensity attributes of sets of colour samples displayed on a computer monitor. The test sets were arranged according to Table 3.1. Five levels of linguistic description were used for each colour attribute. The subjects were not pre-exposed to example graded values or graded representations that might exist for each descriptive grading range. Fifty random colour samples, indicated by "vary" in the Table, were repeated in three sets to comprise a test for each colour attribute: *Hue*, *Saturation*, and *Lightness*,

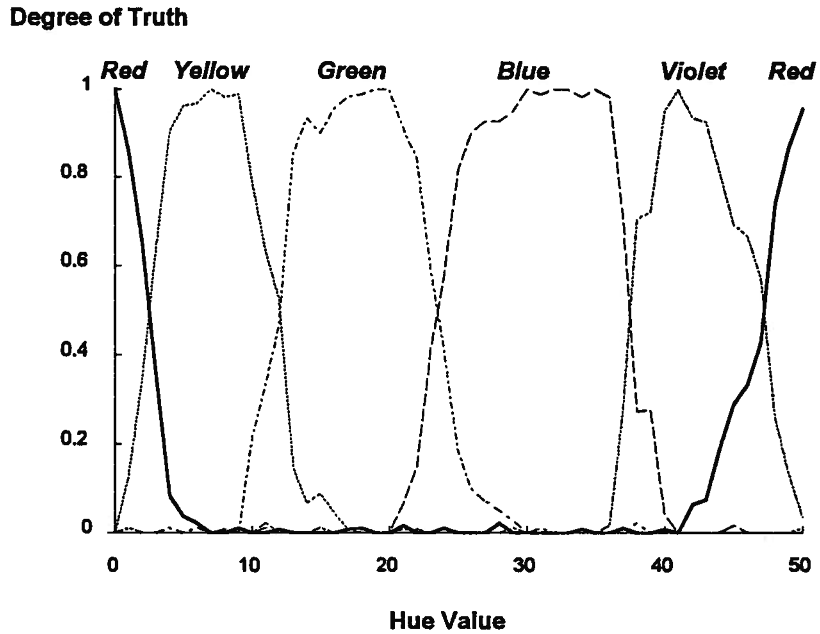
Table 3.1 COLOURIZER Test Organization

	Set 1 (50 samples)			Set 2 (50 samples)			Set 3 (50 samples)	
Hue angle (1)	Saturation radius (2)	Lightness length (3)	Hue angle (4)	Saturation radius (5)	Lightness length (6)	Hue angle (7)	Saturation radius (8)	Lightness length (9)
Hue Test								
vary 0-50	50	25	vary 0-50	50	25	vary 0-50	50	25
Saturation Test								
0 (red)	vary 0-50	25	8 (yellow)	vary 0-50	25	33 (blue)	vary 0-50	25
Lightness Test								
0 (red)	50	vary 0-50	8 (yellow)	50	vary 0-50	33 (blue)	50	vary 0-50

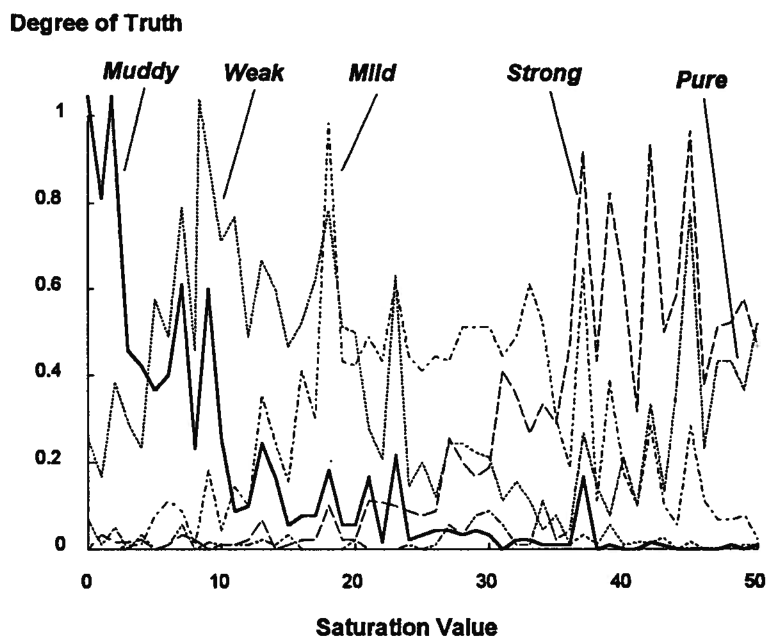
respectively. On completion of each test, COLOURIZER constructed membership/certainty functions using the classed data from each test subject, and posed ten true-false questions to the subject for response about the agreement between a pre-programmed colour sample, and its linguistic description as ascertained from the test subject's membership functions. The students took about 40 minutes to complete the instruction set.

Data files mentioned in the instruction sheet were gathered from the computers for analysis. Each data file captured key membership function points, and raw and relative truth data with respect to dimensionless values ranging from 0 to 50 for the various classifications within *Hue*, *Saturation*, and *Intensity*, respectively. The membership functions from all test subjects for each grading were then compared to assess their agreement. If general agreement was found, then the quantitative measuring parameter had been chosen correctly implying that the subject designers were in general agreement with the definition and subtleties of colour concepts. If poor agreement existed, then perhaps the measuring parameter was at fault and needed adjusting, or test subjects were not assessing the cases accurately.

Again, in the interest of time and the consideration of the suitability goal, it is useful to deal in generalities and thus the data was reduced to simple aggregate averages with no weighting. The data revealed the results in Figure 3.18 plotted from the average of all student classified responses for *Hue*, *Saturation*, and



(Saturation radius 50, Lightness length 25)



(Hue angles 0, 8, 33; Lightness length 25)

Figure 3.18 Average Classified Colour Data - *Hue, Saturation, Intensity*

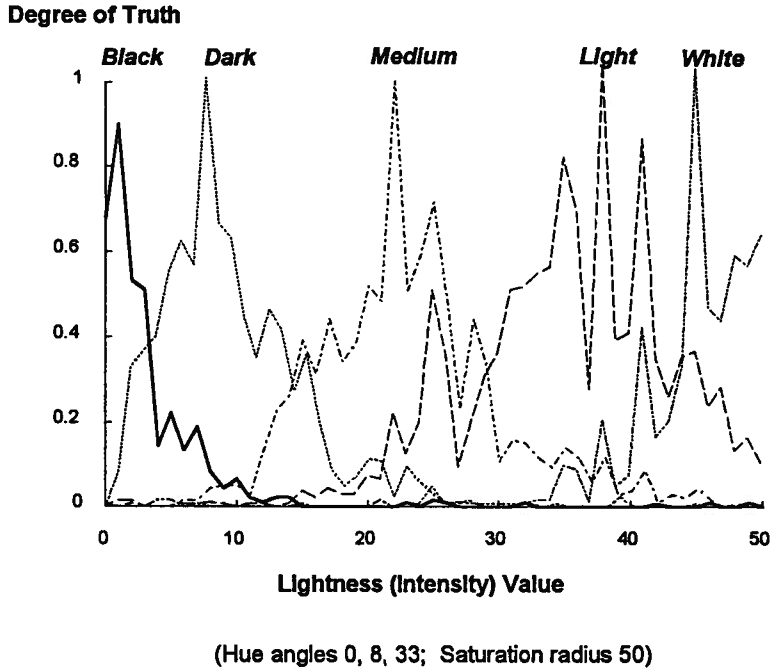
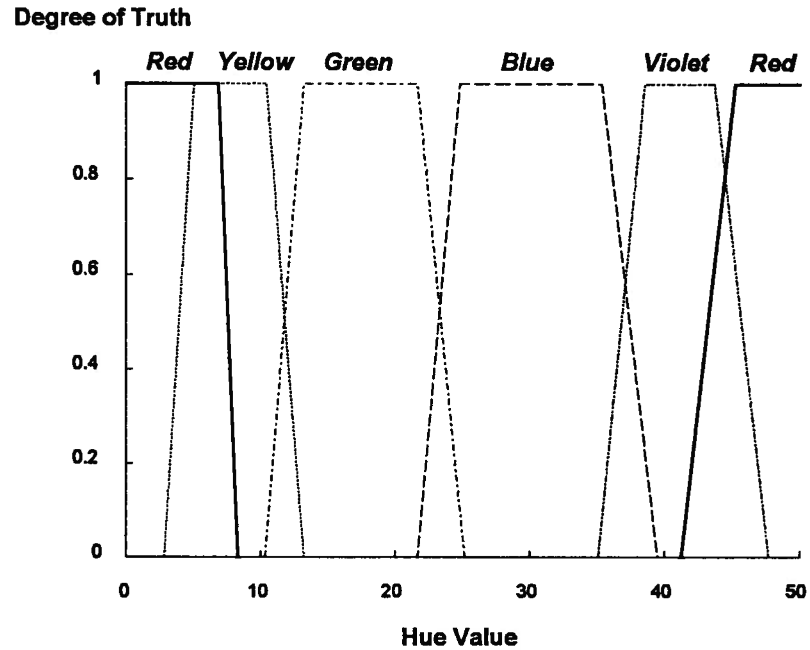


Figure 3.18 *continued*

Intensity corresponding to the respective tests in Table 3.1. Linked points are for illustration purposes only. The scatter of all classified data was visually represented quite well by the averaged relations given above. Of immediate note is that the data in each case is strongly segregated into the five classifications, with little overlap in the odd-sequenced classes (i.e. Red and Green). Thus it would appear that the quantitative measuring model is generally correct, particularly for *Hue*, and *Intensity*. The odd-sequenced overlap in *Saturation* however may be a result of classification judgment error on the part of some students, however it is rather small. This may be as a result of the lack of classification range preconditioning of the subjects before classification took place. The maximums of the peaked classification curves are close to a truth ratio of 1 (at least over 0.9) indicating that the quantitative-qualitative model is indeed believable. The shape of the test data distribution suggests acceptance of the simplified linear tent-shape membership function form for each data classification. Figure 3.19 shows the averaged membership functions



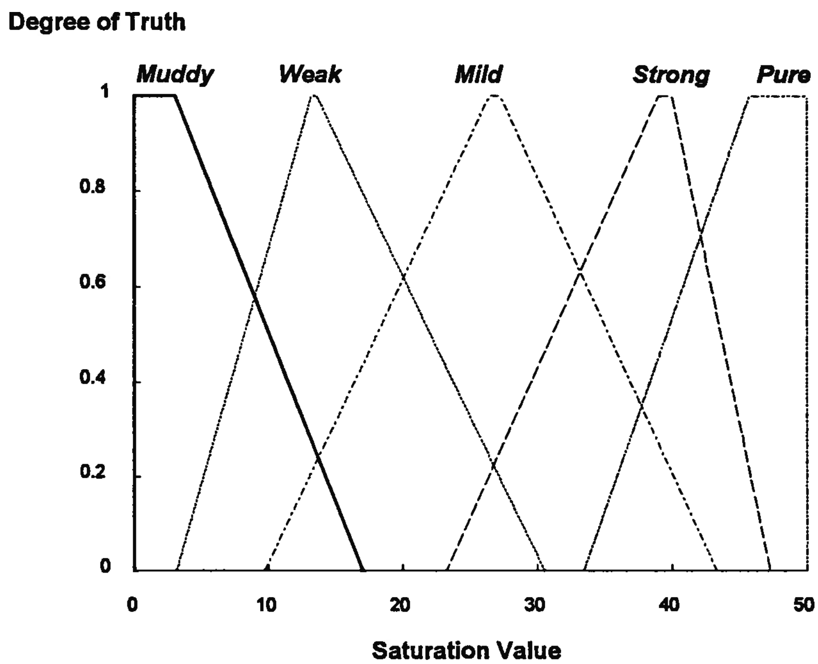


Figure 3.19 Membership Function Fit to Data - *Hue, Saturation, Intensity*

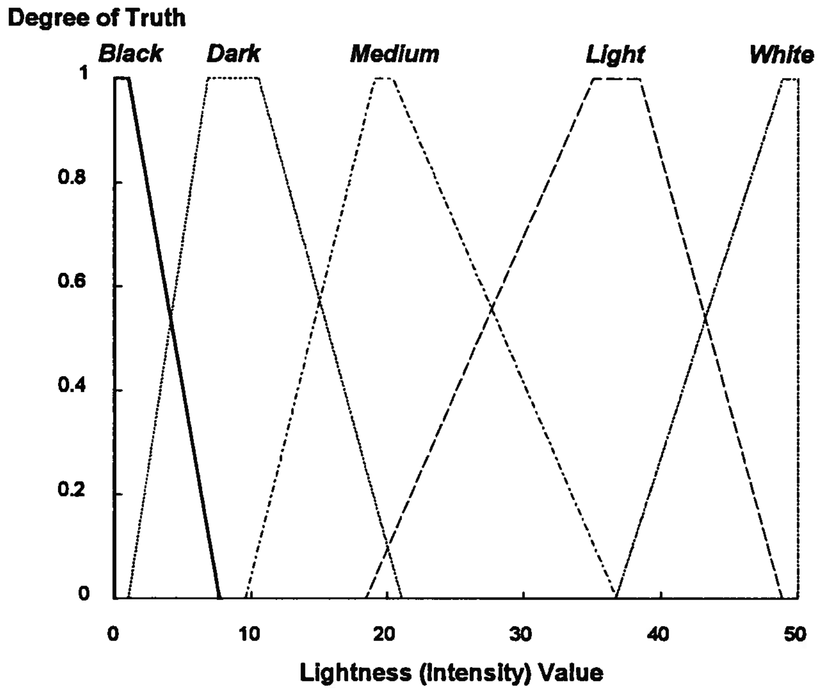


Figure 3.19 *continued*

constructed from the colour description calibration from the 30 test subjects. Again, the membership functions are strongly segregated and take on the familiar tent form. The odd-function overlap in *Saturation* reflects the judgment difficulty experienced by the test subjects due to the lack of classification range preconditioning. This was evident during the test as many test subjects complained of uncertainty of the possible range in meaning of the linguistic descriptions. The exercise was useful in that it showed what happens when a subject is confronted with classifying information into regimes of uncertain defined meaning. As more cases are dealt with, the subject gradually acquired knowledge of the boundaries of the descriptive meaning on the fly. This is reflected in the increasing segregation of the membership functions with increasing case classification. The variance of the membership functions among test subjects was rather small. Also from limited observations, colour blindness did not appear to significantly alter data values or derived membership functions.

Results from the 10-case mini-quiz at the end of the test indicated an average agreement with the membership function-derived descriptions of the colour cases of about 55 percent with a standard deviation of 18 percent. While not spectacular, the low number reflected the poor membership function judgment of *Saturation* due to the test subjects' problems at classifying *Saturation*. This classification problem supports the requirement of a dynamic model in the sense that if more data is added, the membership function points can shift to reflect this acquisition of new knowledge. Over time, the shifts would become progressively minor, reflecting the "experience" influence, hence raising the accuracy of judgments made by membership functions.

3.8 CONCLUSIONS

It appears that an adaptation of fuzzy logic membership functions may be appropriate for bridging the qualitative (often linguistic) and quantitative (often numerical) aspects of architectural/engineering design for a number of valuable reasons:

- dynamic definition of membership functions allows the knowledge base to grow with each new design experience beyond a basic level (the system learns),
- membership function shape flexibility reflects a designer's own intentions which are not constrained by the imposition of another's design values, but yet remain consistent with those of a general population,
- membership functions provide a good translation tool between qualitative and quantitative ill-conditioned information, as well as easy manipulation of that ill-conditioned information.

A study of the proximity of boxes using fuzzy logic membership functions demonstrated a first step in knowledge-based evaluation of a number of architectural/aesthetic qualities, i.e. *colour, texture, shape, size,* that can be used in the "IF" portions of rules for aesthetics. One can continue to create a number of membership functions for each artifact that translate from the physical (quantitative) to the subjective (qualitative), or in reverse. The dynamic membership function approach does not sacrifice the personal taste of the designer, but in fact might help clarify it to the point that design intentions are in general readily understood. The final step to *meaning* from *language* can also be developed in the same manner. Using a smart object representation with appropriate object attributes, one can begin on setting up ways to assess *dominance, balance, unity, etc.* using object attributes in a singular and collective sense that can suggest a design *language* for a collection of objects (the "THEN" side). Again, it is expected that the constructs of these relationships are in themselves ill-conditioned, and will therefore require the use of membership function adaptation as before. Thus, fuzzy set theory in this context can be used to model the ill-conditioned design/analysis process, particularly the subjective side.

The colour study continued to support the same premises made above with respect to using membership functions from fuzzy set theory to link quantitative to qualitative design information. The technique allows for a richness in aesthetic design brought about by the subtle variations possible for a common universal meaning. The colour study reported here supports this idea, as well as suggests a dynamic approach to reflect the gaining of experience for poorly understood classifications. This is particularly important for calibration of the membership functions. Fuzzy logic membership functions are shown to be a simple and effective way to deal with the modeling issues while providing meaningful results. However, a minimum level of calibration is required as the membership functions are only as reliable as the amount of accumulated classification test data used to construct them, reflecting the attitude of learning by experience. The colour study showed that good results can be obtained with small data collections provided the calibrator's knowledge is clear in the possible range of meanings of descriptive linguistics for a certain design quality such as colour. Large calibration data sets are required for attributes where meanings of descriptors are unknown, or unclear. Membership functions from limited observance appear to transcend physical challenges such as partial or total colour blindness, as similar member functions were achieved for normal and physically challenged test subjects alike.

The motivation of the work in this chapter was to explore a means for designers to assess and interact with qualitative and quantitative design issues for an intended artifact design. In particular, the importance of this work is justified in the design of visible structural connections where often the architectural and engineering issues require detailed attention for the artifact to succeed as a whole. Independent consideration often leads to a poor aesthetic, structural failure, cost overruns, poor service performance, and many other problems. It is hoped that this work will provide a path to integrating dynamic subjective issues into what is currently a prescriptive automated design process.

Chapter 4 AESTHETIC CONSIDERATIONS FOR TIMBER CONNECTIONS

4.1 INTRODUCTION

A holistic view of design not only includes the safety and practical issues common to engineering, but also the visual impact and meaning of structure that comprises a work of architecture. A specialty of this thesis in the bridging of the architecture and engineering disciplines is to shed some understanding of how each of the design disciplines deal with the design of connections. Chapter 2 previously, and others to follow, present relevant engineering issues in this context, while this chapter follows the qualitative-quantitative translation discussion of Chapter 3 in dealing with architectural aesthetics and meaning of designed connections in timber. Building on the translation bridge of reasoning in a numerical world as engineers can do; architects, in describing a designed artifact, do so in linguistic terms which leads to a rationalization process that is primarily linguistic. The discussion to follow is developed around describing and reasoning linguistically the complex area of architectural aesthetics for connections.

4.2 BACKGROUND

Bazjanac (1974) reviews models of the architectural design process and some underlying theories of architecture which forms a significant part of the contemporary body of knowledge of architecture. Among these and perhaps the earliest, Vitruvius (1916) Book I, Chapter 2 cites six fundamental principles and three additional that must be satisfied if a work of architecture is to be expected to have quality:

1. *Order* - the selection of modules from the members of the work itself and starting from these individual parts of members, constructing the whole work to correspond,
2. *Arrangement* - the putting of things in their proper places and the elegance of effect which is due to adjustments appropriate to the character of the work,
3. *Eurythmy* - beauty and fitness in the adjustments of the members,
4. *Symmetry* - proper agreement between the members of the work itself, and the relation between the different

parts and the whole general scheme, in accordance with a certain part selected as standard,

5. *Propriety* - that perfection of style which comes when a work is authoritatively constructed on approved principle. It arises from prescription, from usage, or from nature,
6. *Economy* - denotes the proper management of materials and of site, as well as thrifty balancing of cost and common sense in the construction of works,

A1. *Durability* - assured when foundations are carried down to solid ground and materials wisely and liberally selected,

A2 *Convenience* - when the arrangement of the apartments is faultless and presents no hindrance to use, and when each class of building is assigned to its suitable and appropriate exposure, and

A3. *Beauty* - when the appearance of the work is pleasing and in good taste, and when its members are in due proportion according to correct principles of symmetry.

A similar description of qualitative merit for successful design in the more modern sense is presented in Edel (1967) who lists 7 formal traits found in designed objects of attractive appearance:

1. *Rhythm* - a regular occurrence or alteration in design elements,
2. *Dominance* - a part that is most easily perceived through some outstanding difference with the rest of the design to attract attention,
3. *Balance* - a visual equilibrium in design features that may be either symmetrical or asymmetrical,
4. *Transition* - a gradation from one feature to another that may be abrupt or gradual,
5. *Variety* - a diversity or difference in design elements in varying degrees,
6. *Contrast* - an opposition of design elements in varying degrees, and
7. *Unity* - a harmonious relationship of all the design elements woven together.

Using all of the above traits is not necessary in all designs. Satisfaction of two or three traits used together usually is sufficient for a high chance for the aesthetic success of the design. It also easy to overdo an effect.

A tectonic perspective on architectural quality is described by Shack (1989) as a building design quality that provides an appropriate evocative response in the beholder based on the way in which the building was made. This idea of making embodies and supplements the attributes of quantity design from engineering:

- *Strength* - the ability of the structure to safely sustain and resist all imposed loads, stresses, strains, and effects without material failure,
- *Serviceability* - the ability of the structure to deform and perform under load within measurable acceptable limits, and
- *Economy* of means.

However, perhaps more importantly, Prak (1968) states in the context of artistic quality that the *meaning* of art is primarily one of *feeling*:

> ....the works of art are symbols of emotions. The efficient use of emotional meanings in language distinguishes poetry from doggerel. A work of art should *do* something to us - strike a chord. An aesthetic sense demands that the felt emotional importance of the function finds some expression in the architecture. Architectural aesthetics is a more or less coherent system of criteria which are formal and symbolic at the same time. A general aesthetic free from values shows only a pattern.

He further notes that *formal aesthetics*, rooted in psychological structure, deals with proportions, rhythm, repetition, formal cohesion, consistency, or those attributes that can be directly sensed physically through sight, touch, smell, taste, sound, and toxicity. In most cases, these characteristics can be quantified and graded. *Symbolic aesthetics* employs epithets which are heuristic-rooted in the meaning which particular forms have for a particular society at a particular time. Symbolic examples are: *honest*, *truly modern*, *barbarian* vs. *good*, and have been shown to be based in Gestalt psychological laws of perception.

Finally, Prak (1968) claims that symbolism decides which architectural forms will be used and how they will be brought into play by contrasting one against the other. The integration between the elements of architecture into a coherent building is the job of formal aesthetics. The symbolically necessary contrasts are compensated for by closure, repetition, simplicity of overall mass, symmetry, etc. The symbolic aspects of a period defines the area of freedom for its corollary, formal aesthetics.

A work of architectural quality then has, as a primary attribute, the ability to evoke a feeling within the beholder in some way. The artifact is a form of communication between the artist and the beholder. Evocation of beholder feelings is inspired by the artist's skill in developing a particular aesthetic *style* for the artifact, and thus, aesthetic style is an important communication tool.

Edel (1967) tabulates general elements of visual design that contribute to an aesthetic style as: line, value (lightness of darkness of one colour or more), texture (the variation of a surface conditioned by its structure, perceived visually as well as tactually), shape, space, and colour. Features of these elements are described in Table 4.1 for background.

Table 4.1 Elements of Visual Design - Edel (1967)

Element (1)	Types/Aspects (2)	Qualities (3)	Functions (4)
Line		Long or short Thick or thin Curved or straight Even or tapered Continuous or broken Light or dark (Value) Black, white, or any colour Expressive or descriptive Fluid or geometrical Precise or vague	Define outer edges of a shape (outline). Emphasize 3D quality, imply solidity. Shade a form, define a light source (shadow). Show action of gesture (convey feeling of movement). Show space (converging or tapered lines). Form a texture. Set up a linear pattern. Convey information. Elicit a specific response.
Value			Define elements of shape, line, texture, or colours. Show movement (by contrast). Show space. Show changes in shape. Define the intrinsic light or dark of an object. With plane to create illusion of volume or solidity.

continued

Table 4.1 *continued*

Texture	Differences in light or dark due to a *source of light*. Differences due to mixtures of materials having *different values*. Differences due to mixtures of materials having *different colours or combinations of the above*.	Derived from natural or synthetic surface conditions.	Define a shape, line, or value. Help define a colour. Show movement. Show space.
Shape	*Organic*: pertaining to biology (amoebae), living, moving. *Geometric*: pertaining to mathematics. *In-between*: combination of above.	Large or small (Space) Smooth or rough (Texture) Light or dark (Value) Black, white, or any colour Even or tapered (gesture, Space) Derived or invented (subject matter) Flat, suggesting three dimensions, or actually three-dimensional Isolated, adjacent to another shape, or overlapping (proximity) Linear or massive (tectonic or stereotomic)	Emphasize 3D form (shape as an area of shading). Show gesture (direction, or movement). Show space (large and small shapes, overlap, tapered shape). Form a texture. Symbolize an idea or object. Elicit a specific response.
Space	*2D*: L × W *True 3D*: L × W × D *4D*: Space, time concept	Illusion of no depth Illusion of limited depth Illusion of limitless depth	Space is that essential element in which all other elements exist.
Colour	*Hue*: the identity of the colour *Intensity (Saturation):* the brilliance of a colour (purity and strength) *Value (Lightness, Intensity):* the lightness or darkness of a colour.		Create space (advance and recession). Define a shape, line, or texture. Create specific emotional responses and psychological effects.

Although fashions and tastes change, visual design elements and aesthetic principles provide a basis for developing appropriate aesthetic styles for works of architecture.

4.3 AESTHETIC DESCRIPTION OF CONNECTIONS

In studying mechanically fastened connections of structural members, it can be shown that a wide variety of aesthetic styles exist that help convey meanings within the architectural space: from industrial, to ornate, to clean and neat, to brutal, or ugly. The elements of visual design that typically come to bear for connections are: colour, texture, shape, pattern, and scale. A designer can adjust any of these visual elements within ranges to create an evocative style that can diminish or increase the aesthetic effectiveness of the designed artifact. Examples of the range possibilities of these visual elements are shown in Figure 4.1. The figure shows visual design elements separately for hardware and fasteners to illustrate range possibilities of each, but realistically the same visual design elements apply to both. Since structural members are assumed here to be prescribed in terms of size and shape, only colour and texture of the member surfaces remain as visual element variables for members. However, architects are known to change size and shape of members to suit an overall aesthetic if connection aesthetics prove inappropriate or infeasible.

Another important feature of visual design is given by the term *visual mass*. The eyes and attention of a person can operate together similar to the zoom lens of a camera: creating a frame of vision within which contains all objects that draws the attention of the person. Visual mass represents the amount of projected area of an object as seen within the frame of vision of the beholder. An object with a high amount of visual mass tends to dominate the field of vision. If a group of like objects are present within the field of vision, then the visual mass of the collection of objects is the product of the visual mass of one object and the number of objects in the collection. Now the designer can control not only the range of visual element impact of an object, but also how much the visual impact weighs against other objects in the field of vision. The skill in the balancing of visual element range and visual mass of visible structural objects is the expertise required to develop an appropriate aesthetic for the architecture - one that will lead to the designer's intended meaning for the work.

4.4 A PROCESS FOR ADDRESSING CONNECTION AESTHETIC CONCERNS

The development of a process for including connection aesthetics in design should include the following steps:

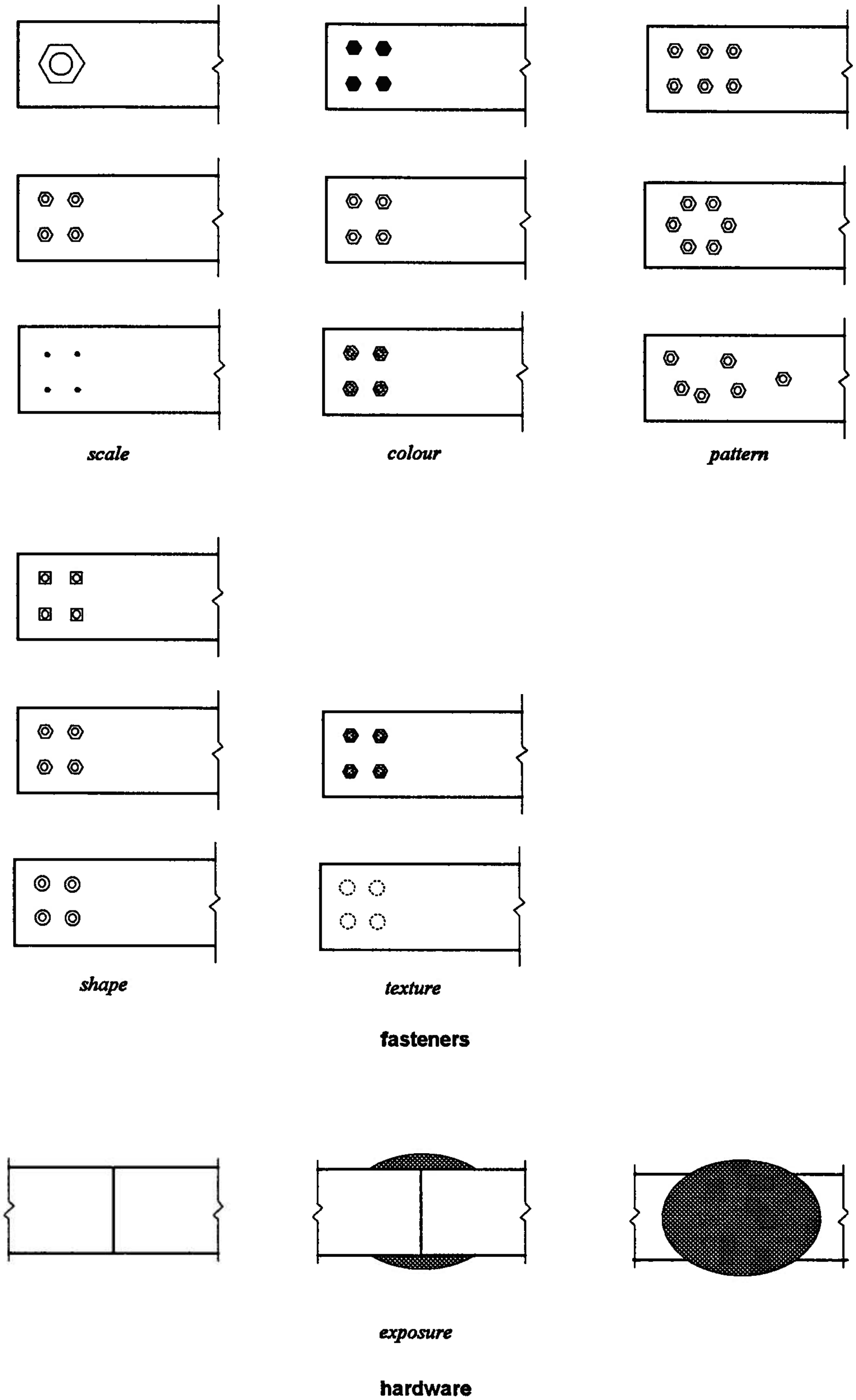


Figure 4.1 Visual Design Elements for Connections

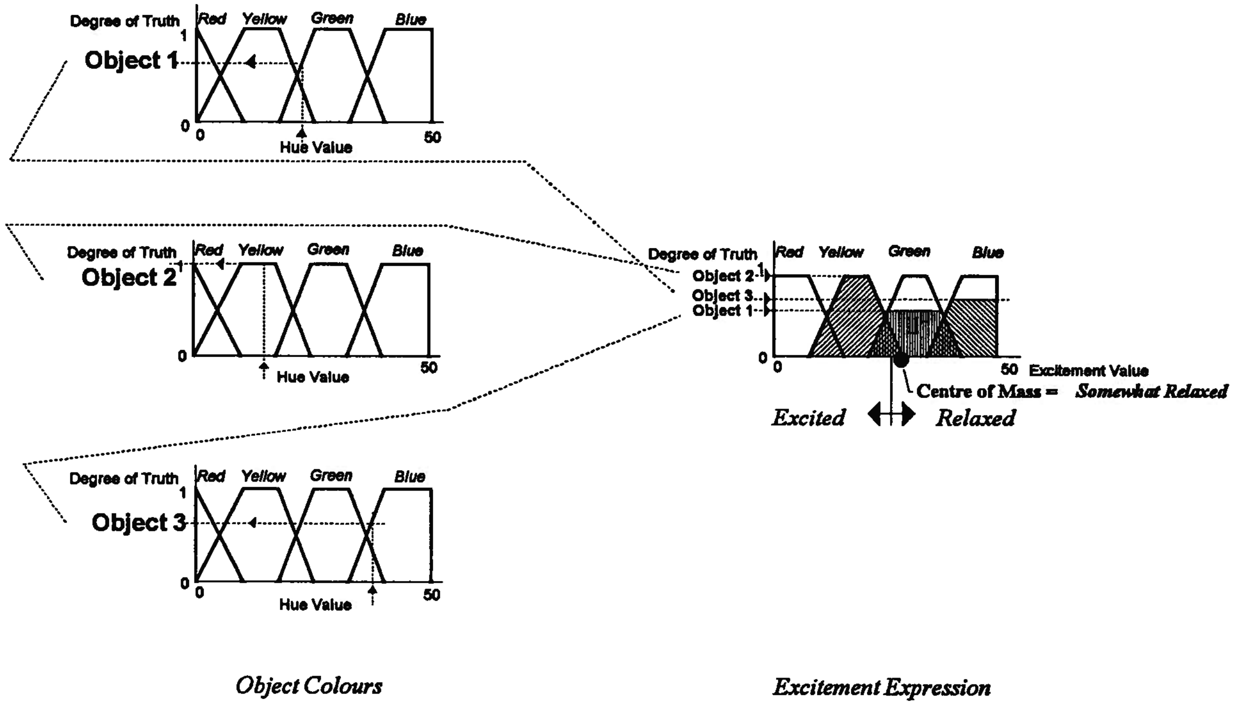
1. translation from the quantitative model space to the qualitative linguistic description space for all visual design elements of objects within a chosen field of vision (to permit linguistic reasoning),
2. an evaluation method for all visual design elements in the context of a design style towards a good aesthetic, and
3. a rating scheme to determine if an appropriate aesthetic has been found for all objects within the chosen field of vision.

The sections to follow deal with each of these steps.

4.4.1 Aesthetic Description and Measurable Physical Reality of Objects

If it is believed that what is seen can be measured, then fuzzy logic membership functions as seen previously form an effective translation tool for the forming of linguistic perception rules. Two studies were undertaken to check the effectiveness of the membership function concept. The PROXIMIZER study detailed in Chapter 3 sought to gauge human perception of the proximity of a pair of boxes. The findings of the study were encouraging in supporting the membership function application idea. A second more elaborate COLOURIZER study also detailed in Chapter 3 sought to gauge human perception of the three attributes of colour (*Hue*, *Saturation*, and *Lightness*) and yielded similar encouraging results. On the basis of these studies, it appeared that fuzzy logic membership functions could be used as an effective perception modeller without any loss of human individuality or uniqueness. This finding forms the basis for translation between what is seen and that is measureable in some way (quantitative) and what is described linguistically (qualitative), which is useful in the writing of linguistic rules that are capable of analyzing aesthetic qualities of objects.

Aside from quantitative-qualitative translation, membership functions can serve another purpose. Object attributes can be tied to inferential psychological responses. A group of such responses can constitute a global "feeling" or expression about the work. Membership functions can be used here to gauge truth of stimulus/response characteristics for a collection of attributes. For example, Figure 4.2 shows colour *Hue* membership function relations for three objects.



IF Object 1 = SomeWhat Green AND Object2 = Definitely Yellow AND Object3 = SomeWhat Blue
THEN Excitement Expression for the group of Objects = SomeWhat Relaxed

Figure 4.2 Aesthetic Meaning Determination for Objects

The colour value of each object is known (say, a *Hue* value from CRT display). At each colour *Hue* value, the degree of truth of a linguistic description of the colour Hue is ascertained from where the value intersects the membership function; the higher the degree of truth, the more likelihood the corresponding linguistic descriptor is the correct descriptor. Knowing the linguistic descriptor of the colour *Hue* of the three objects and their corresponding degrees of truth, it is possible to relate this information to human perception. If membership functions for the degree of truth of linguistic descriptors for colour *Hue* are known for scaled values of emotional feeling from excited to relaxed, the collected colours of all objects can suggest an appropriate emotional response. This is done by overlaying areas under the corresponding object truth values for the corresponding linguistic membership functions. By calculating the centre of mass of the trapezoidal areas (by taking moments) along the emotion scale values, a discrete value for emotion in the numerical range from relaxed to excited results. The numerical emotion scale values are tied to linguistic ranges so that an emotion linguistic can result.

4.4.2 Visual Design Element Valuation

Once linguistic values are determined for an object's visual design elements, and to be useful to a rationalization process for a group of objects, bounds must be placed on the linguistic values so that the values take on some relative meaning. The relations are generally built around the words: *same, different, similar, etc.* The bounds, which could be written in terms of a simple linguistic rule, represent a more broad classification of element values to make the rationalization process easier.

Consider the example of colour *Hue* of an object. *Hue* is represented linguistically in the COLOURIZER study by five colours: red, yellow, green, blue, and violet. These terms in themselves are broad classifications for ranges of these colours. However, the colour classifications relate to each other in formal ways. Artists are familiar with the rule that contrasting (different) colours appear in different halves in the Munsell colour wheel (Isaac 1971: 46), while complementing colours appear within the same half of the colour wheel. The stronger the degree of separation, the greater the contrast. Similarly, the weaker the degree of separation, the closer the complement. Contrasting *Hues* tend to create a tension in the artifact, while complimentary *Hues* tend to unify it. A resort to corresponding *Hue* quantity values may be useful in obtaining colour wheel locations and broad classifications for a group of *Hues*, however, further reasoning would normally be done using the linguistic *Hue* design element values of *contrasting* or *complimentary* for the colour grouping. Another example, colour *lightness*, can be broadly classified in similar fashion into *light*, *neutral*, and *dark*, by reducing the five membership function classifications as previously used in COLOURIZER to three. This reduction can result in a more simplified reasoning process in that fewer values for the design element are involved.

4.4.3 Linguistic Reasoning Computation for Aesthetics

An artifact can be judged on its aesthetic merits based on Edel's (1967) aesthetic traits for a collection of objects. It is useful to implement a rating scheme, and to strike a rule that invokes the axiom:

IF high ratings exist in three or less aesthetic categories THEN aesthetic is good.

For every aesthetic merit rule that fires, a credit is posted towards the respective aesthetic traits rating. Evaluation of the rating scores among the aesthetic traits can indicate the presence of an acceptable aesthetic, its strength, and where the aesthetic strength or weakness lies. A description of the process is as follows. If values for all of the visual design elements of all the objects in the connection are known (i.e. colour, shape, texture, scale, etc.) as shown in the dotted grids of Figure 4.3, then one could evaluate each visual trait of ***all objects*** in the collection according to the general rules for aesthetic traits:

- *Dominance* IF for a visual design element, the majority of objects share the same visual design element value
  THEN the visual design element is dominant and DOMINANCE IS PRESENT

This trait compares visual design element values, not objects. In other words, it is only the visual element that can have dominance.

- *Rhythm* IF the majority of visual design element values of one object are the same majority values as other objects
  THEN RHYTHM IS PRESENT

This trait seeks to find majority patterns among the visual design elements of all objects. The repetition of this pattern in each object constitutes a rhythm.

- *Contrast* IF for a visual design element, objects have two possible values and the values are opposites
  THEN CONTRAST IS PRESENT

This trait looks for opposites in a visual design element among all objects in the collection; the stronger the opposition of the element values, the greater the degree of contrast. In addition, the greater the number of visual design elements in contrast, the greater the degree of contrast is shown by the object collection.

- *Variety* IF for a visual design element, objects have unique values
  THEN VARIETY IS PRESENT

This trait can be considered the opposite to rhythm.

- *Balance* IF the visual design element mass centroid of all objects is in centre of the field of vision
  THEN BALANCE IS PRESENT

This trait presents an interesting modelling challenge in that the field of vision must be formally described. However, the balance trait works closely with the ideal of visible mass and where it is placed relative to the centroid of the field of vision for the object collection. If the centre of visible mass for all objects coincides with the centroid of the field of vision, then balance is present; otherwise, the artifact appears to draw the beholder in the direction of the centre of visible mass.

and, finally:

- *Unity* IF visual design element values for given aesthetic traits are similar among all objects THEN UNITY IS PRESENT.

This trait tries to find a common theme linking all objects to some degree. While rhythm deals with somewhat precise repetition, unity is more encompassing in the general sense.

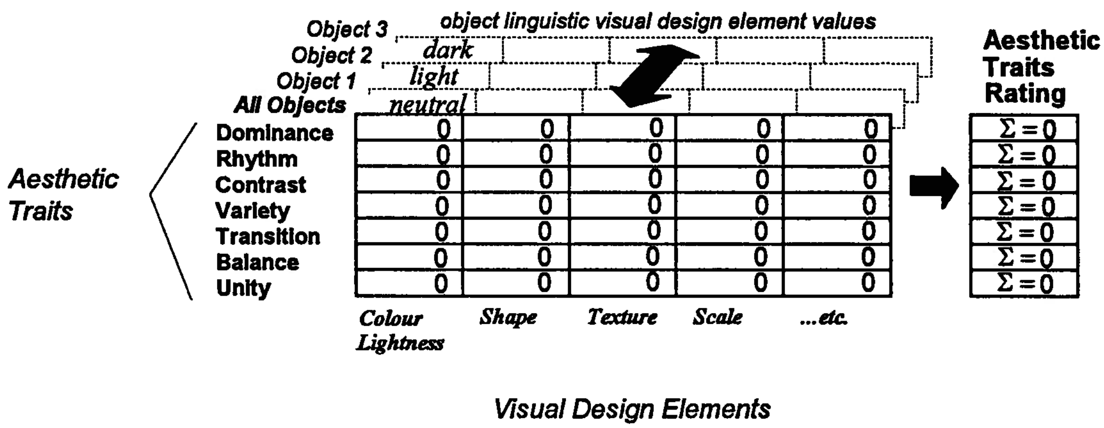


Figure 4.3 Aesthetic Rating of Connection Object Visual Design Elements

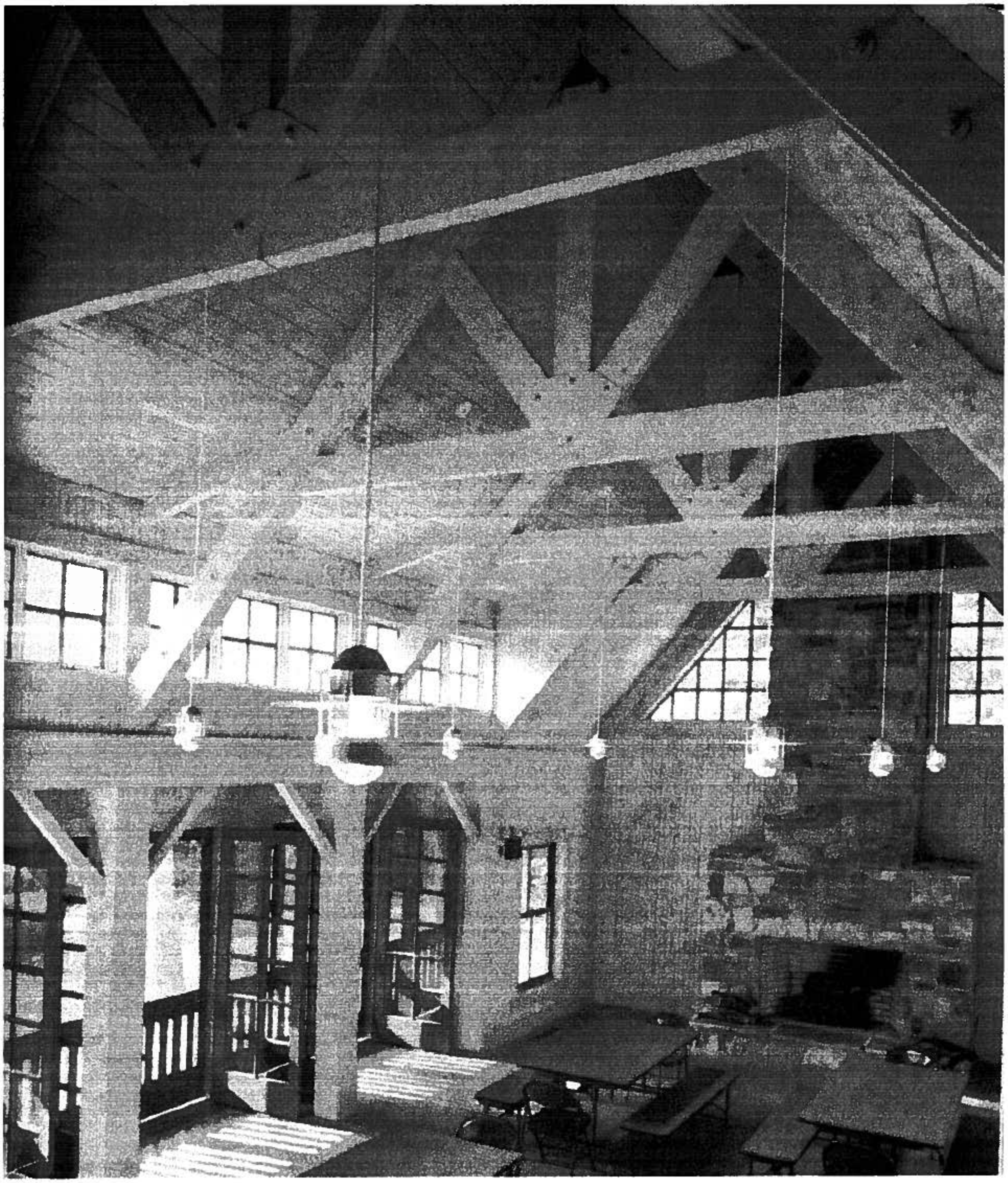
Ratings of a value of 1 in each category for an *all objects* visual design element would be incurred and placed in the solid grid if any rule fires (is met). Finally, the row ratings over all traits for the *all objects* grid are summed in a list to the right, yielding a list of overall ratings for each of the seven aesthetic traits listed above. If three or less of the ratings are close to zero while the remainder are significantly different in value otherwise, then a good aesthetic has been achieved for all objects in the collection. Otherwise, the aesthetic would be somewhat weak. The degree of weakness is reflected in the range in final rating values of the seven aesthetic traits.

4.5 EXAMPLES

It is useful for understanding to demonstrate some portions of the described process of Section 4.4.3 by means of an example of a built design project taken from Canadian Wood Council (1991: 102-103).

Stage 1

Figure 4.4 Truss Connection



Stage 2

Figure 4.4 *continued*

The truss shown in Figure 4.4 shows a number of connections for members. The first stage of Figure 4.4 shows the truss at completion of assembly while the second stage of the Figure shows the truss after application of finishes. The truss assembly consists of mainly three aesthetic groups of objects: members, bolts, and plate hardware.

The linguistic visual design value of colour lightness can be broadly described as *light*, *neutral*, and *dark*. In terms of colour lightness, from visual observation the individual objects can be classed linguistically as: *neutral* members, *light* bolts, and *dark* plate hardware. These values must be scaled for visual mass for the collection as follows:

IF visual mass = large THEN visual design element increases one degree
IF visual mass = medium THEN visual design element remains the same
IF visual mass = small THEN visual design element increases one degree

For the visual mass of members being *medium*, the lightness result is the product of *neutral* and *medium* yielding *neutral*. Similarly, for the visual mass of bolts being *small*, the lightness result is the product of *light* and *small* yielding *neutral*. The logic here is that scaling down negates the colour effect, while scaling up accentuates it. Finally, for the visual mass of plate hardware being *medium*, the lightness result is the product of *dark* and *medium* yielding *dark*. The final scaled values then are: *neutral* members, *neutral* bolts, and *dark* plate hardware.

The next step is to evaluate the lightness result for all the objects in terms of each of the aesthetic traits, again using linguistic rules. In the context of Figure 4.3, this step fills in the first row of the *all objects* grid in Figure 4.5. The dominance rule fires here since the majority of the number of objects in the collection are neutral, bearing in mind that the design element values have already been corrected for visible mass. The firing of the rule sets the number 1 in the (*Dominance*, *Colour Lightness*) cell of the *all objects* grid. The contrast rule also fires because of the existence of two and opposing values of *colour lightness* for all of the objects in the collection, setting a 1 into the (*Contrast*, *Colour Lightness*) cell of the *all objects* grid. The number 1 also appears for balance because of the geometrical symmetry of arrangement and attributes for all the objects within the field of vision. While a challenging conceptual process to model, it should be easy to see from Figure 4.4 that this is so. The remaining cells in the *Colour Lightness* column remain with 0 values either because the associated rules proved false, or were inapplicable. This technique can be followed for other visual design elements and traits, filling out completely the *all objects* grid.

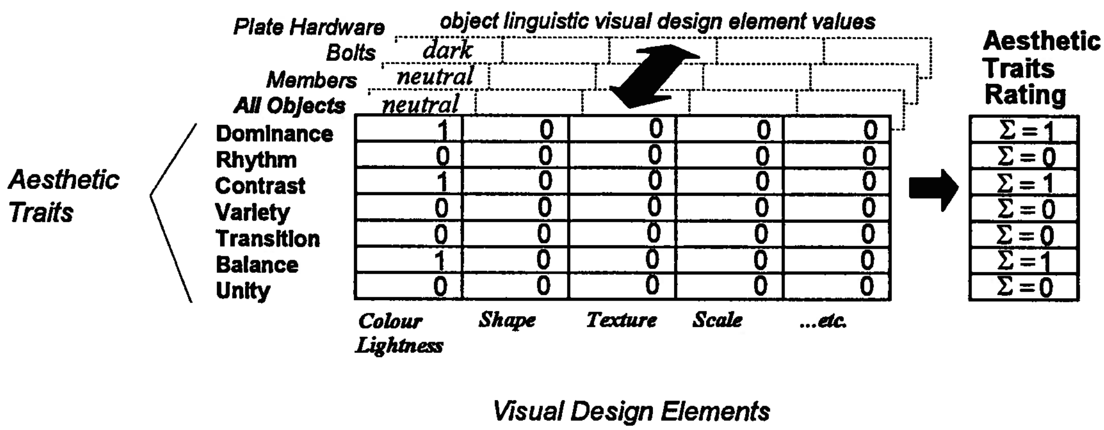


Figure 4.5 Example - Aesthetic Traits Rating for Stage 1

Next, the all object grid rows are summed into the aesthetic traits rating list. By looking at the values in the list, the aesthetic axiom can be checked for validity to see if an appropriate aesthetic was reached. The values in the list are revealing in that the values indicate the relative strengths of the various traits used in the aesthetic strategy, and which could be easily adjusted to make the aesthetic stronger, or weaker. In this way, the designer is fed back useful information on how well the design-in-progress is proceeding towards an attractive end result. This information can prove invaluable equally to designers not skilled in the area of aesthetic design in the context of training, and to skilled designers in terms of their own design experimentation towards new aesthetic possibilities. The particular combination of the values in the *all objects* trait/visual design element grid can be stored under an appropriate aesthetic expression label that can be recalled and reused in another design (or portion of the same design) consisting of similar objects. In this way, perhaps an appropriate expression linked to the meaning of the designer's intentions, can be reused in other portions or fields of vision of the designed artifact.

The second stage of Figure 4.4 features a slightly different aesthetic through the designer's choice of finishes. In this case, the same colour has been used throughout for all structural elements. Using the same collective object terminology as for the first stage; in terms of colour lightness from visual observation, the individual objects can be classed linguistically as: *light* members, *light* bolts, and *light* plate hardware. These values must be scaled for visual mass for the collection as follows:

IF visual mass = large THEN visual design element increases one degree
IF visual mass = medium THEN visual design element remains the same
IF visual mass = small THEN visual design element increases one degree

For the visual mass of members being *medium*, the lightness result is the product of *light* and *medium* yielding *neutral*. Similarly, for the visual mass of bolts being *small*, the lightness result is the product of *light* and *small* yielding *neutral*. Finally, for the visual mass of plate hardware being *medium*, the lightness result is the product of *light* and *medium* yielding *neutral*. The final scaled values then are: *neutral* members, *neutral* bolts, and *neutral* plate hardware.

Again, the next step is to evaluate the lightness result for all the objects in terms of each of the aesthetic traits, again using linguistic rules. This step fills in the first row of the *all objects* grid in Figure 4.6.

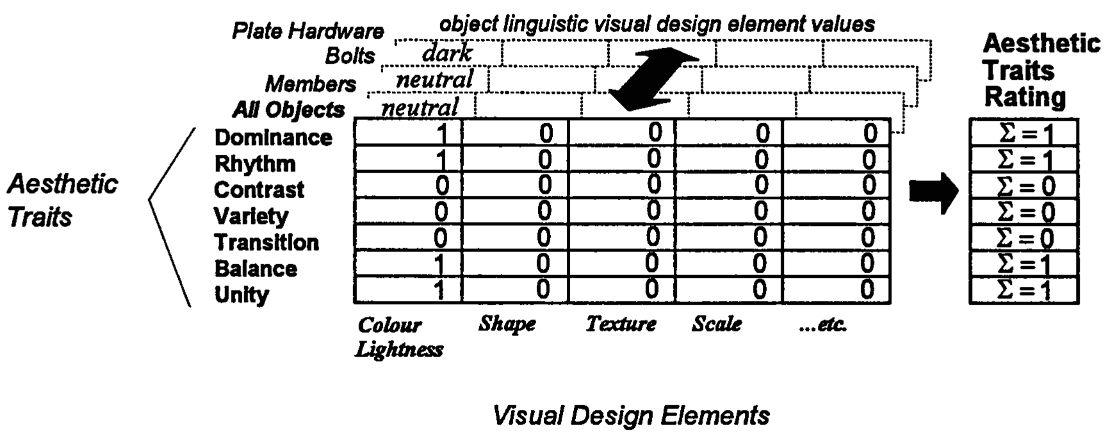


Figure 4.6 Example - Aesthetic Traits Rating for Stage 2

The dominance rule fires here since the majority of the number of objects in the collection are neutral, bearing in mind that the design element values have already been corrected for visible mass. The firing of the rule sets the number 1 in the (*Dominance, Colour Lightness*) cell of the *all objects* grid. The rhythm rule fires since the colour lightness for all the objects is the same value. However, the contrast rule doesn't fire this time because of the preceding reason. The number 1 appears for balance because of the geometrical symmetry of arrangement and attributes for all the objects within the field of vision. The number 1 appears for unity insofar as rhythm has been noted for one visual design element in the object collection. The remaining cells in the *Colour Lightness* column remain with 0 values either because the associated rules proved false, or were inapplicable. Again, this technique can be followed for other visual design elements and traits, filling out completely the *all objects* grid.

Next, the all object grid rows are summed into the aesthetic traits rating list. By looking at the values in the list, the aesthetic axiom can be checked for validity to see if an appropriate aesthetic was reached. The values in the list are revealing in that the values indicate the relative strengths of the various traits used in the aesthetic strategy, and which could be easily adjusted to make the aesthetic stronger, or weaker. In this way, the designer is fed back useful information on how well the design-in-progress is proceeding towards an attractive end result. So far in this evaluation of Stage 2; according to the aesthetic traits rating list, four aesthetic traits share equal strength which suggests that the aesthetic used in Stage 2 has weakened with the same-colour painting of all objects in the truss. To make the aesthetic stronger, the designer hopefully would (and does) use the remaining visual design elements to improve the aesthetic trait rating list values for this design. In this manner, this information can prove invaluable equally to designers not skilled in the area of aesthetic design in the context of training, and to skilled designers in terms of their own design experimentation towards new aesthetic possibilities.

4.6 SUMMARY

A holistic view of design not only includes the safety and practical issues common to engineering, but also the visual impact and meaning of structure that comprises a work of architecture. A specialty of this thesis in the bridging of the architecture and engineering disciplines is to shed some understanding of how each of the design disciplines deal with the design of connections. This chapter follows the qualitative-quantitative translation discussion of Chapter 3 in dealing with architectural aesthetics and meaning of designed connections in timber. Building on the translation bridge of reasoning in a numerical world as engineers can do; architects, in describing a designed artifact, do so in linguistic terms which leads to a rationalization process that is primarily linguistic. A brief background in the area of aesthetics as applied to structural connections was given. A methodology was presented for including connection aesthetics in design that:

1. performs translation from the quantitative model space to the qualitative linguistic description space for all visual design elements of objects within a chosen field of vision (to permit linguistic reasoning),
2. provides an evaluation method for all visual design elements in the context of a design style that leads to a good aesthetic, and
3. uses a rating scheme to determine if an appropriate aesthetic has been found for all objects within the chosen field of vision.

A simple real-life example was discussed to highlight the features of the methodology. It is believed that this approach will prove helpful to engineers and architects alike in guiding them to appropriate aesthetic design for connections.

Chapter 5 OBJECT-ORIENTED REPRESENTATION WITH ANALYSIS ENGINE SYSTEM INTERFACE

5.1 INTRODUCTION

A knowledge-based expert systems approach can be a viable means of efficiently producing a smart design tool for use on a personal computer by an architect or engineer. Types of fasteners and their behaviour in different geometric configurations and wood materials has been well documented in various design codes and application manuals throughout the world and provides an "expert" knowledge base on which to draw for establishing facts and decision rules for timber connection design. Collecting, evaluating, and developing this information into a coherent and comprehensive set of facts and rules, and combining it with additional rules from aesthetics and architecture forms the expert input for the knowledge-based expert system.

A timber connection can be broken into the parts shown in Figure 5.1. For design, the adequacy of the

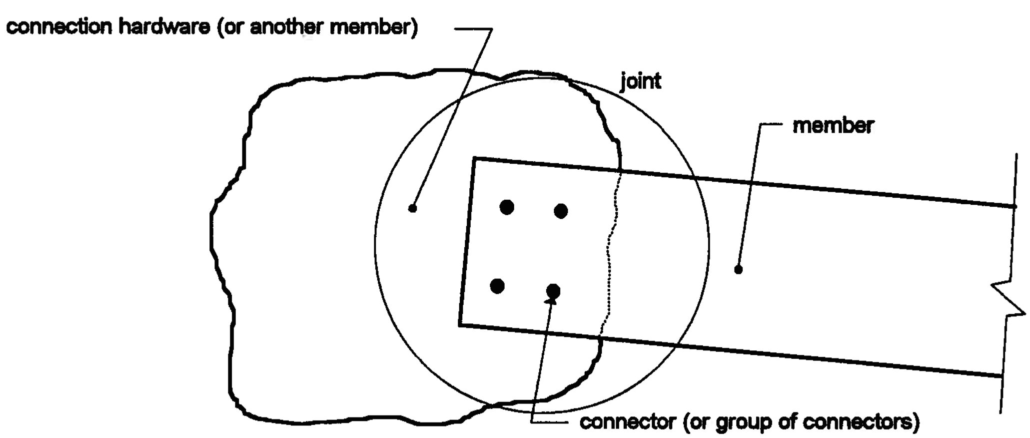


Figure 5.1 Timber Connection Parts

connection structurally and aesthetically is of interest. From knowledge of member sizes, force/moments imparted by a member to the joint, member materials and condition, service conditions, and aesthetic criteria; a simple design logic can be formed:

1. If the joint is multi-member:
   (a) choose basic connection hardware to be used and connector based on aesthetics, and
   (b) break into member connection components.
2. For each member component:
   (a) assume a connector layout pattern and size based on aesthetics, and
   (b) from engineering, check connector capacity (if inadequate, suggest connector re-sizing) and pattern spacing (adjust and re-check if inadequate), and check wood member and hardware capacity.
3. Design connection hardware to accommodate member joint geometry, connector spacing and edge limits, and structural strength capacity.
4. Re-evaluate Step 3. based on desired assumption of Step 1.(a) and revise if necessary.
5. Output design on screen and display relevant design performance information.

In order to test and evaluate the feasibility of the expert system approach for timber connection design software, a pilot project was completed early in the research. The pilot project, TimberCon (Taylor 1992), consisted of development of a small expert system with the aid of an expert system developer's shell (KAPPA-PC) to design a simple two-member butt joint timber connection. Variables considered in the connection included:

- applied action type, magnitude and direction,
- member sizes and material,
- fastener types and placement patterns (initially a limited variety),
- connection hardware (wood or metal plates in addition to bolt or full penetration dowel-type fasteners), and
- aesthetics (hidden fasteners/hardware versus exposed fasteners/hardware in varying degrees of each).

A pleasing graphical user interface was developed to prompt for initial design constrain information, alert the user of potential design problems, and provide graphical line drawing output of the connection design. The basis of the connection design algorithms was the current issue of CAN/CSA 086.1 Engineering Design in Wood (Limit States Design), and the aesthetics portion built upon very simple concepts from the field of architectural aesthetics. Object-oriented programming was used throughout to model the connection (the knowledge base). Rules were written to guide the design process and test the validity of the connection solution using the expert system shell's inference engine, while high level language programming engines were developed to perform analytical calculations for the graphical display of the design. A final critical assessment of the approach as a valid expert system design solution strategy was made considering ease of expansion of the pilot project to include user enhancements (graphical input), engineering analysis engine improvements, graphics display engine upgrading, and enlargement of the architectural and engineering timber design knowledge base so that the expert system could include other connection types, materials, fasteners, geometries, and a wider array of architectural issues.

5.2 COMPUTER MODEL IMPLEMENTATION

An object-oriented approach was used to model the connection and design world, and is shown in the KAPPA-PC implementation of Figure 5.2. Most of the objects are represented well by their object names. The structural elements are composed of member objects and hardware objects as seen in the *Joint* object tree. The *Engineering* object tree holds the link calls to the respective analysis engines of typical parts, and the *Tectonics* and *Graphics* object trees store visibility design requirements and graphics object data and drawing methods, respectively.

A window interface with drop-down menu command boxes was implemented to provide communication between the system and the user. The command structure was constructed in reflection of the overall logic of use of the system. Here, the user is given choice of a number of connection components for a design from a list of parts. Figure 5.3 shows a window used to describe a member to TimberCon and how it is to be connected. As the user chooses components, the list of choices would become more constrained to guide the user in a practical selection. The overall design logic appears in Figure 5.4.

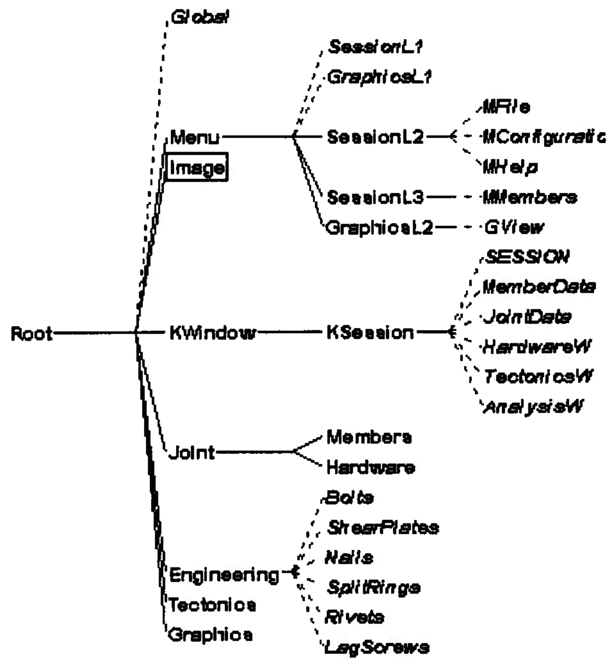


Figure 5.2 TimberCon Object Structure in KAPPA-PC

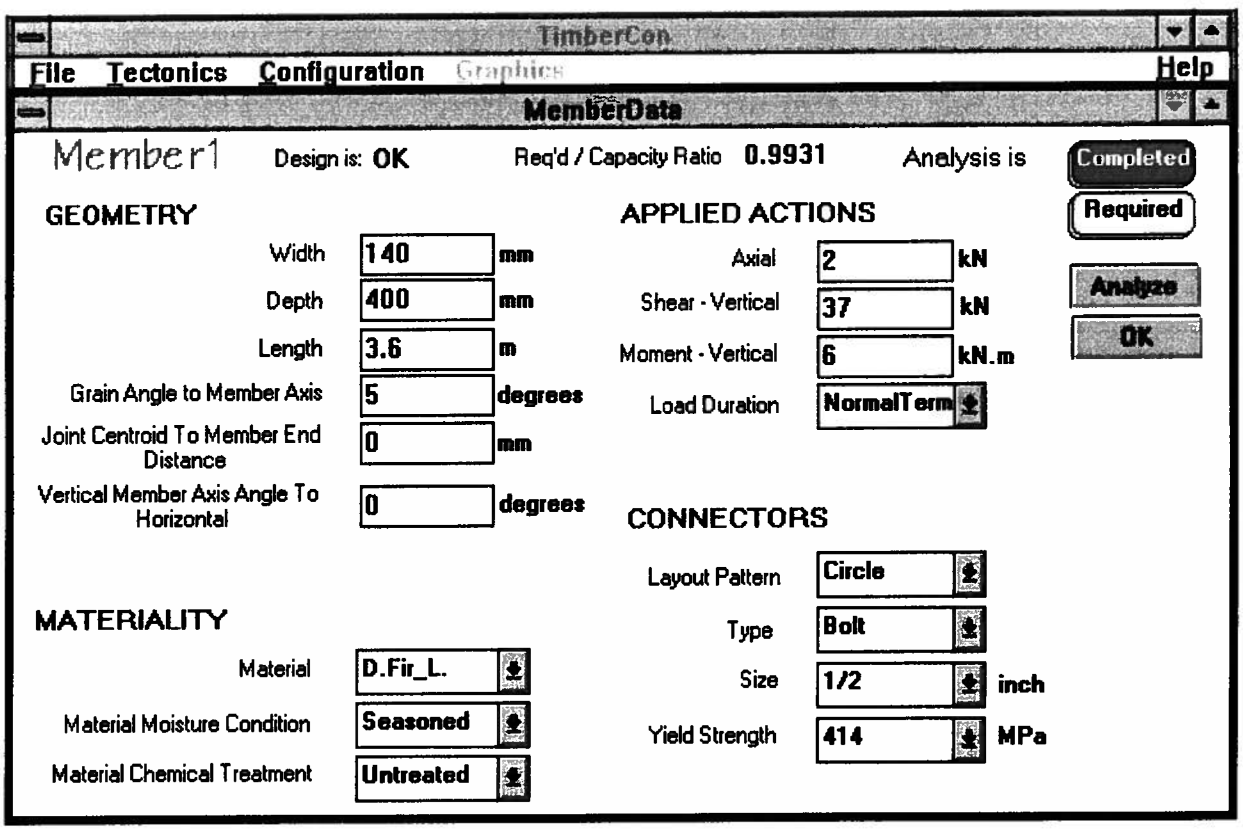


Figure 5.3 TimberCon Member Data Entry Window

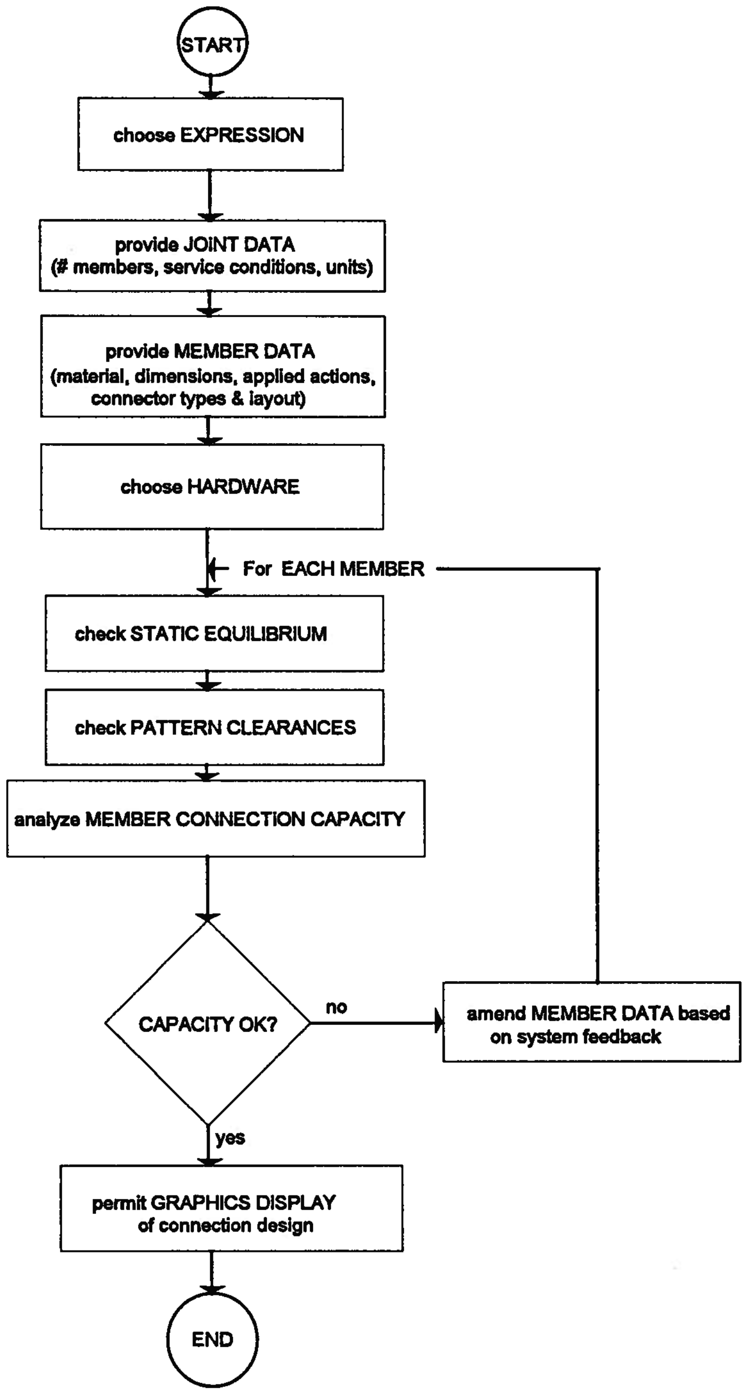


Figure 5.4 Design Logic for TimberCon

5.3 ANALYSIS ENGINE IMPLEMENTATION

The analysis engine was composed of three parts: checking fastener clearances, determining fastener shear resistance capacities, and determining the resistance capacity performance of the group of connectors fastening a particular member. In the pilot study, the analysis engine was developed for bolt fasteners in particular due to the commonplace nature of their usage. At this point in the design, all connector centroid coordinates and member vertices components would be known and calculated from architectural considerations, such as:

- articulation - connection hardware and/or fasteners hidden or exposed,
- connector group pattern - row, grid, circular, etc.,
- connector size - large, small, and
- connector quantity - few, many.

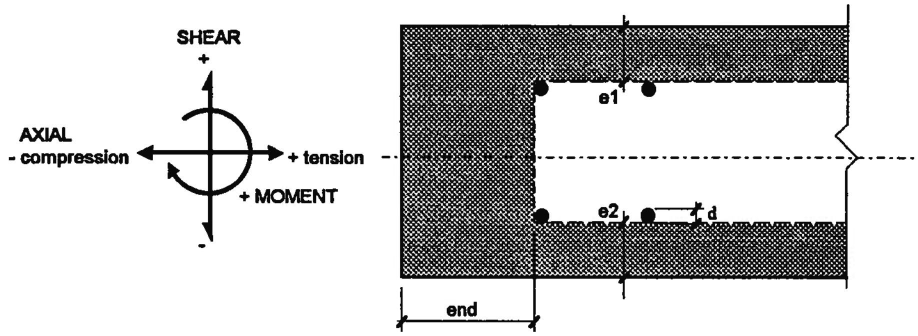
For example, connector coordinates can be easily determined from the following rules knowing the pattern description and the number of connectors N:

IF	SINGLE	AND	$N=1$	THEN	$(x_1,y_1)=(0,0)$
IF	VERT_ROW	AND	$N\geq 2$	THEN	$(x,y)_i=\left(0,\left[\frac{-l}{2}+\left(\frac{i-1}{N-1}\right)\right]\right)\ldots i=1,N$
IF	HORZ_ROW	AND	$N\geq 2$	THEN	$(x,y)_i=\left(\left[\frac{-l}{2}+\left(\frac{i-1}{N-1}\right)\right],0\right)\ldots i=1,N$
IF	RECTANGLE	AND	$N=4$	THEN	$(x_i,y_j)=\left(\left[\frac{-b}{2}+\left(\frac{i-1}{n-1}\right)b\right],\left[\frac{-h}{2}+\left(\frac{j-1}{n-1}\right)h\right]\right)\ldots i=1,n;j=1$
IF	GRID	AND	$N>4$	THEN	$(x_i,y_j)=\left(\left[\frac{-b}{2}+\left(\frac{i-1}{m-1}\right)b\right],\left[\frac{-h}{2}+\left(\frac{j-1}{n-1}\right)h\right]\right)\ldots i=1,m;j=1,n;n=2$
IF	CIRCLE	AND	$N\geq 5$	THEN	$(x,y)_i=\left(r\cos\left(\frac{(i-1)\cdot 360}{N}\right),r\sin\left(\frac{(i-1)\cdot 360}{N}\right)\right)\ldots i=1,N$

where N is the number of fasteners in the group, x and y are corresponding connector coordinates, l is the length of a row, b is the length of the grid along the member axis, h is the length of the grid perpendicular the member axis, r is the circular connector group radius, m is the number of member-axis-perpendicular rows in the grid, and n is the number of member-axis-parallel rows in the grid. The connector group centroid is set to be always at (0,0).

5.3.1 Fastener Clearances

Fastener to wood material edge distances, and interior inter-fastener spacings are of concern here. Rules for edge distance clearances determined from the bolt diameter have been arrayed as a function of applied action and clearance parameter, as in Figure 5.5.



Parameter	Axial +	Axial -	Shear +	Shear -	Moment
end	7 d	4 d, min 50 mm	4 d, min 50 mm	4 d, min 50 mm	7 d
e1	1.5 d	1.5 d	4 d	1.5 d	4 d
e2	1.5 d	1.5 d	1.5 d	4 d	4 d

d = bolt shank diameter in mm

Figure 5.5 Fastener Edge and End Clearances

5.3.2 Fastener Shear Resistance

For this study, the lateral shear capacity N_r at an angle to the wood grain θ was determined for bolts using the yield theory equations of Canada Wood Council (1990). These equations determine a group of bolts' lateral shear strength in wood parallel to the grain P_r and perpendicular to the grain Q_r respectively, and are given as:

[5.1] $$N_r = \frac{P_r Q_r}{P_r \sin^2\theta + Q_r \cos^2\theta}$$

where $P_r = \phi P_u n_s n_f J_F$,

$Q_r = \phi Q_u n_s n_f J_R$,

n_s = number of shear planes across the bolt (2 normally),

n_f= number of fasteners in a group (this was always taken as 1 in the spreadsheet since the lateral shear capacity of only 1 bolt is of interest for comparison to applied forces at fastener),

$J_F = J_G J_L J_R$

where $J_G = 0.33\left(\frac{l}{d}\right)^{0.5}\left(\frac{s}{d}\right)^{0.2} N^{-0.3} \not> 1.0$ (factor for up to 12 bolts in a row),

$J_L = 7d \rightarrow 0.75; 10d \rightarrow 1.0$ (factor for loaded end distance),

J_R = 1.0 (1 row or 1 bolt/row), 0.8 (2 rows with 2 or more bolts per row),

0.6 (3 rows with 2 or more bolts per row), (factor for number of rows),

l = member thickness (mm);

d = bolt diameter (mm),

s = in-row spacing (mm) (minimum $4d$),

N = number of bolts in row,

$\phi = 0.7$ (performance factor),

$P_u = p_u K_D K_{SF} K_T$ (kN),

$Q_u = q_u K_D K_{SF} K_T$ (kN),

where K_D, K_{SF}, K_T = wood material modification factors for load duration, service condition, and fire treatment,

p_u and q_u = the least of (N per bolt):

$$F_1 d^2 \frac{l_1}{d},\quad F_1 d^2 \sqrt{\frac{2}{3}\frac{f_2 f_y}{(f_1+f_2)f_1}},\quad F_1 d^2\left(\frac{1}{2}\frac{f_2 l_2}{f_1 d}\right),$$

$$F_1 d^2\left(\sqrt{\frac{1}{6}\frac{f_2 f_y}{(f_1+f_2)f_1}}+\frac{1}{5}\frac{f_2 l_2}{f_1 d}\right),\ \text{and}\ F_1 d^2\left(\frac{1}{5}\left[\frac{l_1}{d}+\frac{f_2 l_2}{f_1 d}\right]\right),$$

where $F_1 = 0.8 f_1$,

d = bolt diameter (mm),

f_y = bolt yield strength (414 MPa A307 bolt),

$f = 0.088(1-0.01d)\,\rho_k$,

ρ_k = characteristic wood density (kg/m^3):

378	D.Fir-L.
302	Hem-Fir
278	SPF
262	Northern Species

for steel side plates (thickness = 6 mm):

f_1 = bearing strength of steel side plate (574 MPa typical),

f_2 = embedding strength of main member (MPa)

$$= \frac{f}{2.3\sin^2\theta_2 + \cos^2\theta_2},$$

l_1 = 6 mm (side plate thickness, mm),

l_2 = main member thickness (mm),

θ_2 = direction of load angle to main member grain (0° for p_u, 90° for q_u).

for wood side plates:

f_1 = embedding strength of side member (MPa)

$$= \frac{f}{2.3\sin^2\theta_1 + \cos^2\theta_1},$$

f_2 = embedding strength of main member (MPa)

$$= \frac{f}{2.3\sin^2\theta_2 + \cos^2\theta_2},$$

l_1 = side member thickness (mm),

l_2 = main member thickness (mm),

θ_1 = direction of load angle to side member grain (0° for p_u, 0° for q_u),

θ_2 = direction of load angle to main member grain (0° for p_u, 90° for q_u).

The above determination of fastener lateral shear strength was adapted into spreadsheet form as part of the connection group strength determination for bolt fasteners. Determination of the fastener group capacity completes the spreadsheet, for which the method is presented next.

5.3.3 Fastener Group Strength Capacity

The structural behaviour of a group of fasteners in a timber connection is not completely fully understood as reviewed by Taylor (1991) and continues to be an interesting research problem. Stalnaker and Harris (1989) present a rational simplified approach to estimate the capacity of a group of fasteners in a timber joint. The method presented herein is similar with a few variations. The fastener group strength capacity procedure begins with knowledge of the fastener location and group centroid coordinates relative to the member axis, and force/moment values at the joint. It is desired to find the applied resultant shear force applied laterally to each fastener shank recognizing the variable bearing material behaviour due to grain of the wood. Further, the maximum applied resultant shear force is needed so that a comparison to the fastener resistance as determined from code equations can be made. The problem can be constructed as in Figure 5.6.

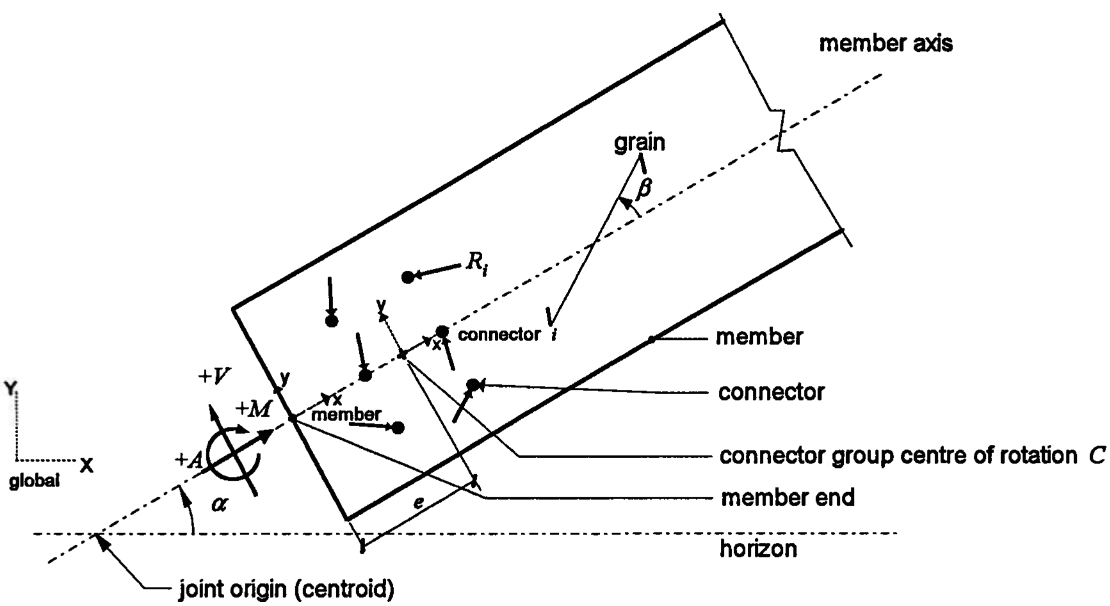


Figure 5.6 Connector Group Analysis Notation

The origin of the connector location coordinates is chosen to be at the geometric centre of the connector group *C*, which also coincides with the centre of rotation of the connectors in the group (i.e. no displacement). The first step is to move the external applied joint forces to the connector group centre of rotation *C*, and convert external applied joint forces to align with the member grain angle using the sign convention of Figure 5.6:

$$\begin{Bmatrix} V_\beta \\ A_\beta \\ M_C \end{Bmatrix} = \begin{bmatrix} \sin\beta & -\cos\beta & 0 \\ -\cos\beta & -\sin\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} A \\ V \\ M+eV \end{Bmatrix} \quad [5.2]$$

This rotation through the grain angle β aligns the applied axial force A_β and the applied shear force V_β parallel and perpendicular to the grain, respectively. The applied moment to the connector group M_C is a summation of the external applied moment at the member end node M, and the product of the applied shear force V and it's eccentricity e taken as the distance from the centroid of the connector group and the end of the member. The relationships of all the force components on each fastener are shown in Figure 5.7.

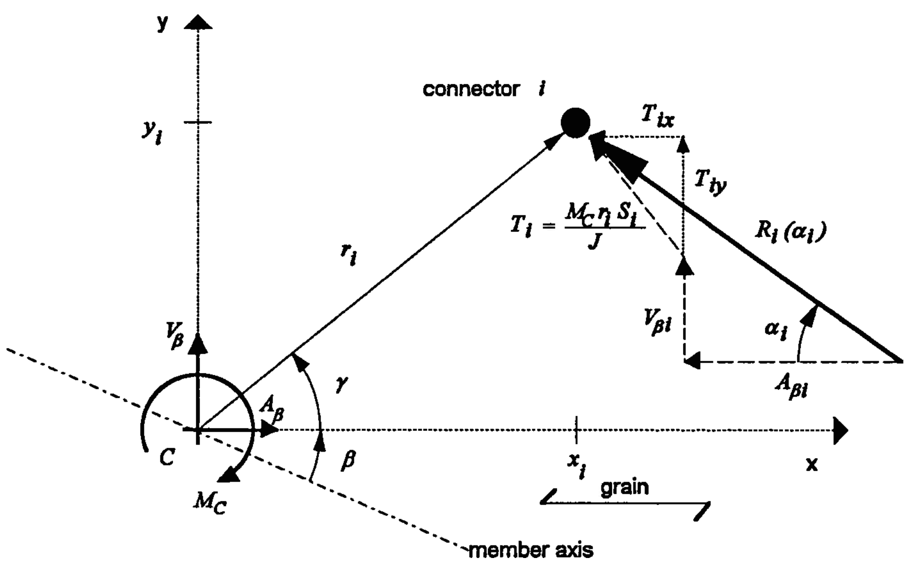


Figure 5.7 Connector Force Components

The moment, acting alone, rotates the connector plate about the centroid of the fastener group *C*. As it rotates, the side member pushes each connector with a force *T*. Force *T* applied to each connector by the moment is in a direction normal to the radius *r* from that connector to the centre of rotation *C*. If it is assumed that the connector plate does not deform severely, the distance each fastener moves is proportional to the radial distance *r*. Thus, if each fastener moment resistance force in the wood *T* is proportional to the distance moved by each fastener, then each *T* is proportional to each *r* respectively. *T* is also proportional to the wood material stiffness that deforms to permit each fastener to move. This stiffness cannot be defined precisely but is assumed here to be proportional to the allowable connector shear load *S*. This is a reasonable assumption since allowable loads for dowel-type connectors are based on slip rather than ultimate strength. Hence, in a relative way for two connectors:

[5.3] $$\frac{T_1}{T_2}=\frac{r_1S_1}{r_2S_2} \text{ or } T_2=T_1\frac{r_2S_2}{r_1S_1}.$$

The sum of the moments of all the *T*-forces about the centre of rotation is equal to the applied moment M_C, thus:

[5.4] $$M_C=\sum_{i=1}^{n}T_ir_i=T_1r_1+T_2r_2+\cdots+T_nr_n$$

where *n* is the number of fasteners in the fastener group.

Now, if the first term of [5.4] is multiplied by $\frac{r_1S_1}{r_1S_1}$, and a relative expression similar to [5.3] is substituted in the remaining terms, one obtains:

[5.5] $$M_C=T_1r_1\frac{r_1S_1}{r_1S_1}+T_1\frac{r_2S_2}{r_1S_1}r_2+\cdots+T_1\frac{r_nS_n}{r_1S_1}r_n=\frac{T_1}{r_1S_1}\sum_{i=1}^{n}r_i^2S_i$$

where the summation term is the polar moment of inertia *J* of the allowable shear loads *S* about the centre of rotation *C* (the fastener group centroid). Rewriting [5.5] gives $T_1=\frac{M_Cr_1S_1}{J}$, or in general for any other bolt force *i*:

[5.6] $$T_i=\frac{M_Cr_iS_i}{J}$$

where $J = \sum_{j=1}^{n} r_j^2 S_j \ldots j = 1, n.$

Each force (due to moment) T computed will be in the direction normal to the radial distance r for each fastener. Hence:

[5.7] $$\begin{Bmatrix} T_{ix} \\ T_{iy} \end{Bmatrix} = \begin{Bmatrix} T_i \sin\gamma \\ T_i \cos\gamma \end{Bmatrix} = \begin{Bmatrix} T_i\left(\dfrac{-y_i}{r_i}\right) \\ T_i \dfrac{x_i}{r_i} \end{Bmatrix} = \begin{Bmatrix} \dfrac{-M_C S_i y_i}{J} \\ \dfrac{M_C S_i x_i}{J} \end{Bmatrix}.$$

The applied axial force A_β and applied shear force V_β aligned with the wood grain angle β are assumed to be divided equally among all fasteners in the group n. Finally, the resultant forces at each connector can be written by summing the vector components in the x and y directions, respectively as:

[5.8] $$\begin{Bmatrix} R_{ix} \\ R_{iy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{A_\alpha}{n} + T_{ix} \\ \dfrac{V_\alpha}{n} + T_{iy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{-A\cos\beta}{n} - \dfrac{V\sin\beta}{n} - \dfrac{M_C S_i y_i}{J} \\ \dfrac{A\sin\beta}{n} - \dfrac{V\cos\beta}{n} + \dfrac{M_C S_i x_i}{J} \end{Bmatrix}$$

where n is the number of fasteners in the group. The radial angle to the grain α_i of each resultant fastener resisting force R_i is:

[5.9] $$\alpha_i = \arctan\left(\frac{R_{iy}}{R_{ix}}\right).$$

Now knowing the correct angle of the resultant force to the grain of the wood for each fastener, the initial estimated strengths S can be revised using the new angle value and the procedure repeated until α_i converges.

The steps necessary in estimating the strength capacity of a group of fasteners in wood can be summarized as follows:

1. Determine fastener resistance values P_r, Q_r and N_r from yield theory equations, and estimate each fastener resultant resistance force angle to the wood grain θ as: $\theta_i = \arcsin\left(\dfrac{x_i}{r_i}\right)$.

2. Calculate the estimated resistance force (due to moment) T of each fastener at the corresponding force-to-grain angle θ_i using Hankinson's formula: $T_i = \dfrac{P_r Q_r}{P_r \sin^2\theta_i + Q_r \cos^2\theta_i}$.

3. Calculate relative strengths S of each fastener based on: $S_i = \dfrac{T_i}{T_1}$.

4. Calculate the polar moment of inertia J according to: $J = \sum_{i=1}^{n} r_i^2 S_i$.

5. Calculate values in [5.7], [5.8], and [5.9] to obtain a more accurate determination of the fastener resultant resistance force angle to the grain of the wood for each fastener α_i.

6. Let $\theta_i = \alpha_i$ for each fastener and recycle through Steps 2 to 6 until θ_i and α_i converge, yielding corresponding values of the resultant resisting force R at each fastener . The capacity of the connection group is reached when any fastener's lateral shear capacity $N_r(\theta)$ first equates to the value of the resultant resisting force at the fastener $R(\theta)$.

The above procedure was adapted to a spreadsheet approach in TimberCon in which the spreadsheet can determine if the connector group satisfies strength design and the amount by which the fastener group is over- or under-designed for strength. The spreadsheet was created for bolted joints only and a portion of it appears in Figure 5.8. The input parameters are: connection configuration data such as fastener and group centroid location coordinates relative to the member axis, wood material strength and size criteria, fastener sizes, number of shear planes; and applied force and moment values at the end of the member containing the fastener group. The sheet returns joint capacity information to TimberCon through cells H13 and H14. TimberCon is linked to the spreadsheet through corresponding object slots and spreadsheet cells respectively.

5.4 GRAPHICS IMPLEMENTATION

A primary attempt was made at modelling the graphics interface in the context of a volumetric object representation within the confines of KAPPA's very limited graphics capabilities. The limited palette of drawing functions constrained the environment to working with line segments, hence wire-frame representations. Line segment end point data was initially stored based closely to the form of the object represented. For example in Figure 5.9, beam members were represented as a 3D wire frame box, thus data was stored for standard locations of the box's 8 vertices. Many other planar or rectangular prismatic elements shared this representation, such as plate hardware. Similarly, bolt heads and nuts were represented as extruded hexagons requiring the storage of 12 standard location vertices data. Bolt shanks were simply represented by their centrelines because of KAPPA's difficulty in drawing extruded circles or ellipses. Each of the standard shapes could be altered in form by scaling, translating or rotating the standard forms according to input or calculated data. New coordinate values are calculated, which are fed to the drawing method within each object so it could draw itself in the view demanded

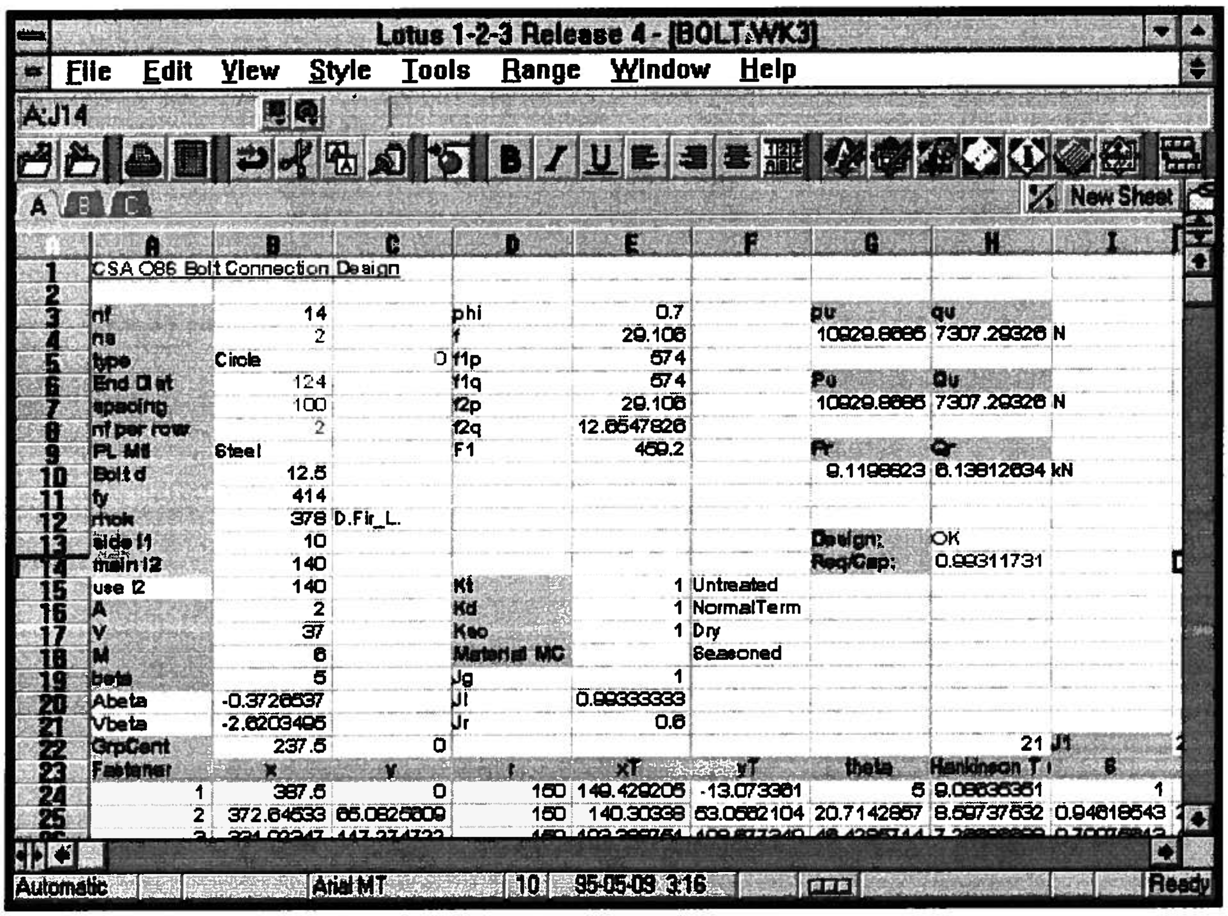
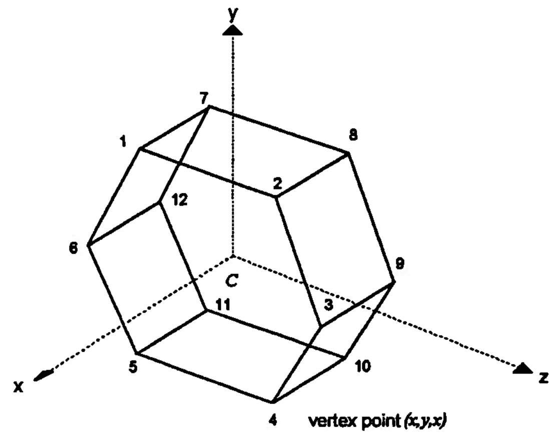
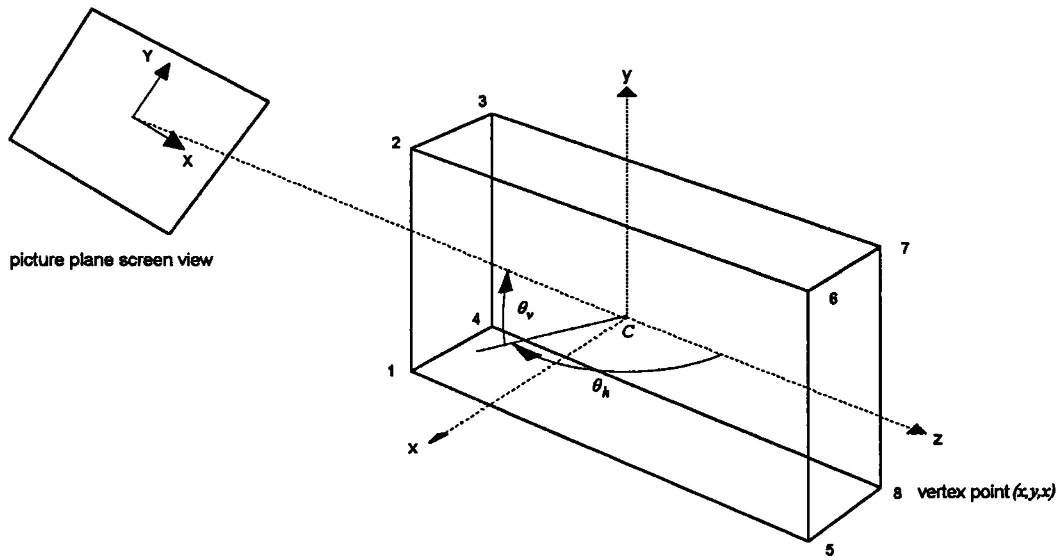


Figure 5.8 TimberCon Linked Spreadsheet Analysis Engine



nut / bolt head



box

Conversion operations for 3D object vertex point manipulation into a 2D viewing plane:

$$\begin{Bmatrix} X \\ Y \end{Bmatrix} = \begin{bmatrix} \cos\theta_h & 0 & \sin\theta_h \\ \sin\theta_v \sin\theta_h & \cos\theta_v & -\sin\theta_v \cos\theta_h \end{bmatrix} \begin{Bmatrix} S(x+T_x) \\ S(y+T_y) \\ S(z+T_z) \end{Bmatrix}$$

where:

where:	S	= scale up/down factor of 3D object,
	T_x	= translation of 3D object in x direction,
	T_y	= translation of 3D object in y direction,
	T_z	= translation of 3D object in z direction,
	X,Y	= 2D point coordinates in picture plane,
	x,y,z	= 3D vertex point coordinates of 3D object,
	θ_h	= horizontal viewing angle to 3D object *z*-axis in *x*-*z* plane, and
	θ_v	= vertical viewing angle to 3D object *x*-*z* plane.

Figure 5.9 Standard Volumetric Wire Frame Objects

by the user. Many of the mathematical routines available to this do work can be found in computer graphics texts such as Kalay(1989) and Glassner(1990). In this manner, the object which holds capacity information is also the same object that can represent itself to the user. This bond provides a powerful facility for critical response of an object using some evaluation engine when the object is viewed or manipulated in some way by a user. In TimberCon's case, the limited drawing representation capabilities of KAPPA proved to be a major impediment to more fully exploiting this point, resulting in the limited solution representation of Figure 5.10.

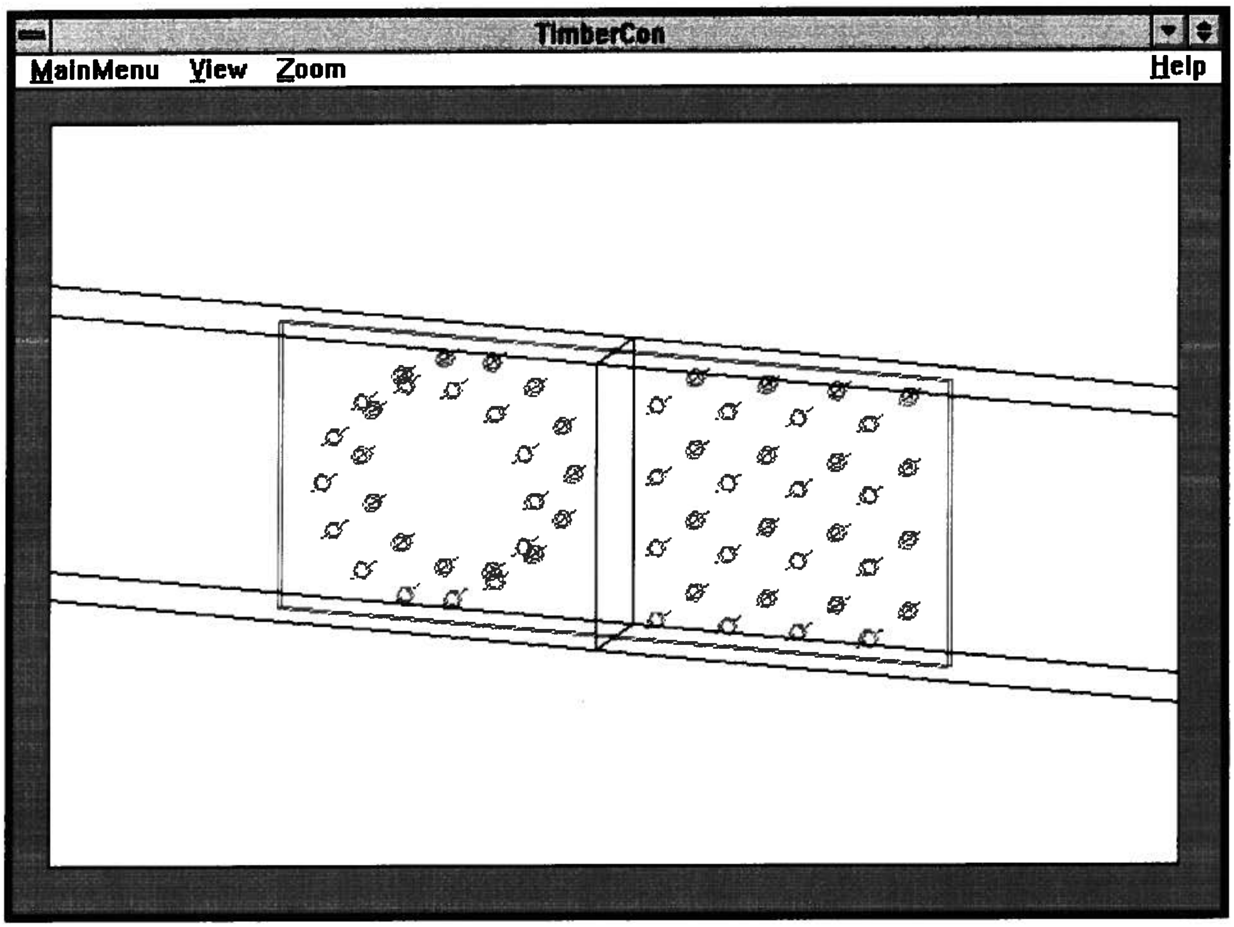


Figure 5.10 TimberCon Graphical Solution Window

5.5 CONCLUSIONS

This study focused on designing a butt joint connection for two members using a plate(s) Transferor. The application is capable of specifying detailed fastener sizing and layout information (for dowel-type fasteners only) and displaying the end result in 3D wire frame. The graphics capabilities are extremely slow due to KAPPA's limited graphics command palette. The analysis engine was constructed in a Lotus-for-Windows Version 3 spreadsheet that proved difficult to work in iteration mode when linked to KAPPA using dynamic data exchange (DDE). Knowledge for the analysis engine followed a relative strength approach as found in Stalnaker and Harris (1989) and used CSA O86 connector resistance formulae and clearance rules. CSA S16 criteria was used to determine minimum plate thickness. The problem of iteration using DDE and high software overhead proved difficult for a spreadsheet approach here, suggesting the use of simple high-level language program executables to service as analysis engines, and temporary data files to pass information between the KAPPA program and the analysis engine executable.

Chapter 6 AUTOMATED DESIGN IMPLEMENTATION FOR CONNECTIONS

6.1 INTRODUCTION

Architectural designers often and customarily create physical models out of cardboard, wood, or other materials to explore design intentions and develop a design solution to higher states. A computer system that captures this hands-on activity in the same manner would make the transition from physical media to electronic media less difficult. Essentially, the designer is modeling object representations of the real components of the design; representations that not only have physically descriptive attributes, but functional and semantic attributes as well. Computer software that is developed around an object-oriented approach (Forde 1989, Coyne *et al.* 1990, Vermeulen 1993) can easily capture in a logical way a snapshot of the physical world it represents. Hence the system should use an object-oriented philosophy to represent the real world that the designer is exploring, much in the way the designer experiences the real life experience.

The real world allegory approach taken for design in the previous chapter can be modelled using an object-oriented approach in a computer. Computer objects can replicate real world counterparts easily through attributes, thus simplifying the modelling process in light of the significant volume of information that must be dealt with. Computer objects can model their own behavioural response (methods), as well as their relative response with other objects through functions and rules. Coupled with a reasoning engine to deal with rules, a computerized design environment can be implemented. Ideally, a fully featured computer CAD package that offers parametric modelling of CAD objects with appropriate function/rule-writing abilities and rule reasoning engines would be the ideal as the user interface, artistic and graphic object manipulation engines are already in place, and implementation of the preceding knowledge would be relatively straight forward.

However, due to unavailability of such a tool at the time for implementing the ideas presented in the previous chapter, an object-oriented expert system shell KAPPA-PC (Intellicorp 1992) was used. While KAPPA-

PC's modelling and reasoning ability suited the modelling portion of the work, the product was unsuitable for vivid graphic representation and manipulation due to its extremely limited graphics function palette. As a result, this reduced the graphics representation to simple symbolic representations and linguistics. However, it is believed that full-featured parametric CAD is the appropriate technology for further development of this work. This chapter highlights features of the implementation.

6.1.1 Basic Features of Development and Implementation

From the outset, it was desirable for practical reasons to use a high level software language as much as possible in the development of the system, which led to the previously mentioned constraints. The system was targeted to the DOS/Windows™ environment running on 80486 CPU's. KAPPA-PC provided a graphically oriented environment for easily defining classes and instances of objects and their data attributes and methods. Object-oriented programming differs from conventional procedural programming in that procedural programming dwells upon what is done to the data, and object-oriented programming dwells upon the interactions of objects. An object, shown in Figure 6.1, consists of one or more pieces of data (referred to as *slots* in the shell using the notation

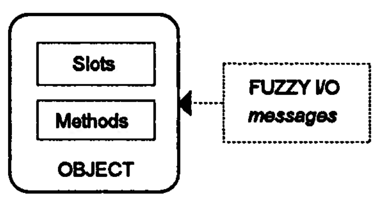


Figure 6.1 Object Representation

object:slot = slotvalue, a set of messages to which it will respond, and a set of procedures (called *methods*) that are invoked by the messages. An object would be a class of objects, or it could be a subclass of a class, or it could be an instance of a subclass or class. The expert system shell provides for the inner working of the objects (the real point of using objects), inheritance, and polymorphism. Inheritance means that all subclasses and instances of a class inherit the characteristics of the parent class. Polymorphism means that different objects can be sent the same message and the objects will respond appropriately in their own way. The shell also supports true message passing between objects, and the messages can be anything from activation of a method within the target object to the activation of the inference engine. Facilities for building rule-based knowledge bases using object-oriented techniques are provided in the shell. The inference engine supports backward and forward chaining, and can be monitored through a rule trace facility. Functions to create dialogue boxes, get input for particular slots, run external applications, use Dynamic Data Exchange (DDE) to exchange data in the background with applications that use it, and calls to directly manipulate ASCII files and start external programs are provided. A session window can be dressed up with sliders, pictures, text, buttons, and other interactive objects that some users find useful. Allowable input values can be included as part of the slot definition. Graphics can be linked to data values in application objects and dynamically updated as the program runs.

The expert system developer's shell works with objects with integrated slots (attributes) and methods, functions, rules, and reasoning engines which appears to the developer as the screen of Figure 6.2. This screen

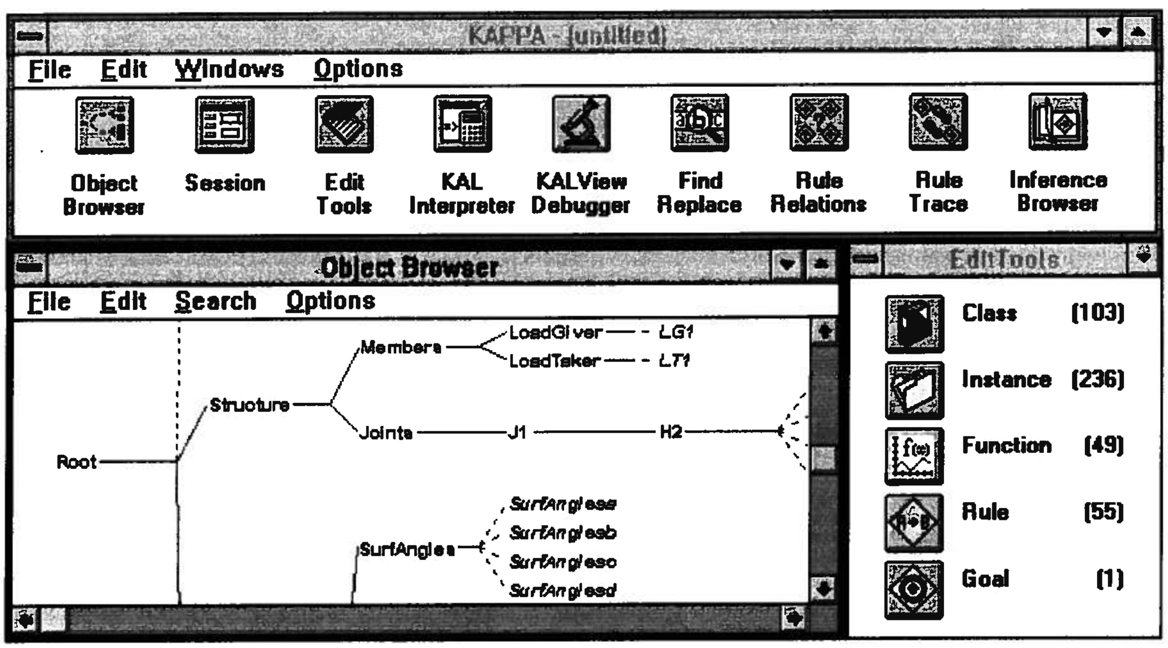


Figure 6.2 KAPPA-PC's Developer's Shell

provides all the tools and monitoring features for software development. The Object Browser contains a graphical representation of all the objects used in the application, while the Edit Tools box provides buttons that bring up editing screens for the various instances, functions, rules, etc. By selecting an object in the Browser or using one of the editing screens, the user can provide information or programmed functions written in a C-like language called *KAL* (KAPPA Application Language) to the application under development. The developed KAPPA application is stored as an ASCII text file that contains C-like KAL language code, which can subsequently be compiled and linked with the KAPPA kernel to produce an executable .DLL or .exe file. Alternately, the application can be stored as a binary file which can be run using the limited run-time form of KAPPA-PC. The runtime version can remain in the background behind the application such that the user only sees and interacts with the application, and not the KAPPA machinery.

An object tree shown in the Object Browser models the world as in Figure 6.3 where solid lines connect

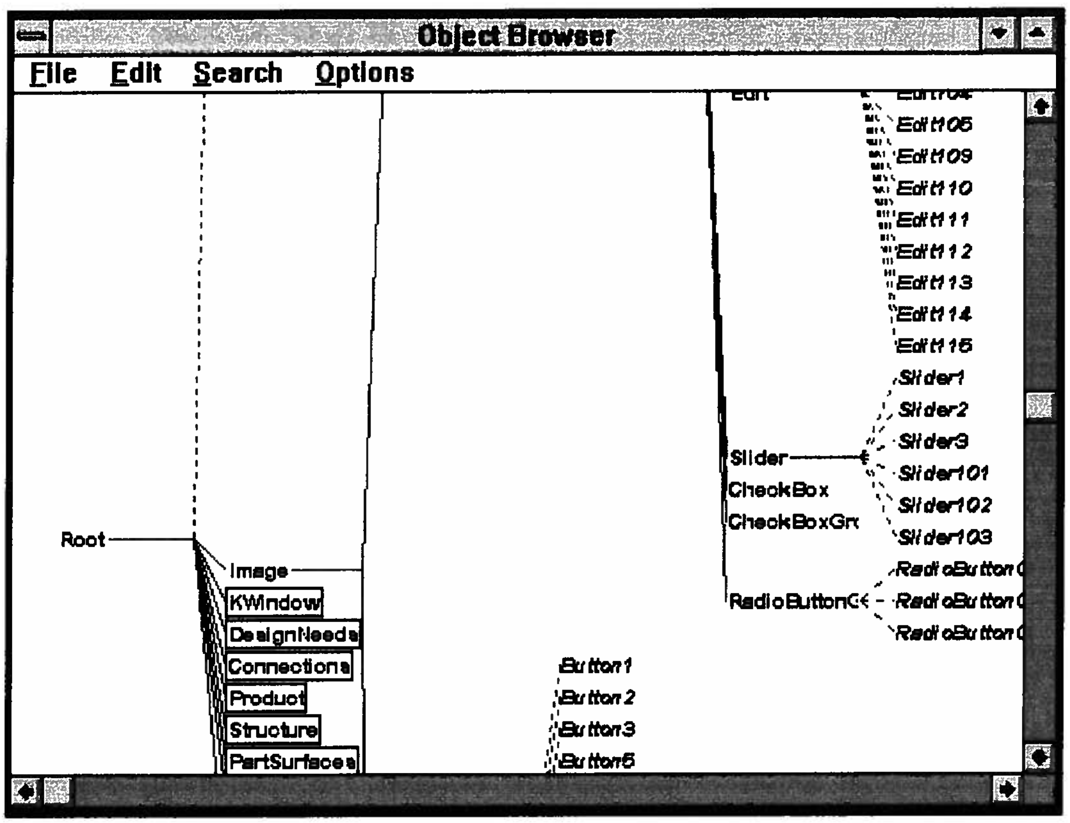


Figure 6.3 Object Browser

class objects and broken lines extend to instance objects of a class object. Object properties are given in *slots*, which are accessed as mentioned by selecting an object in the Browser, to call up a screen like Figure 6.4, in this

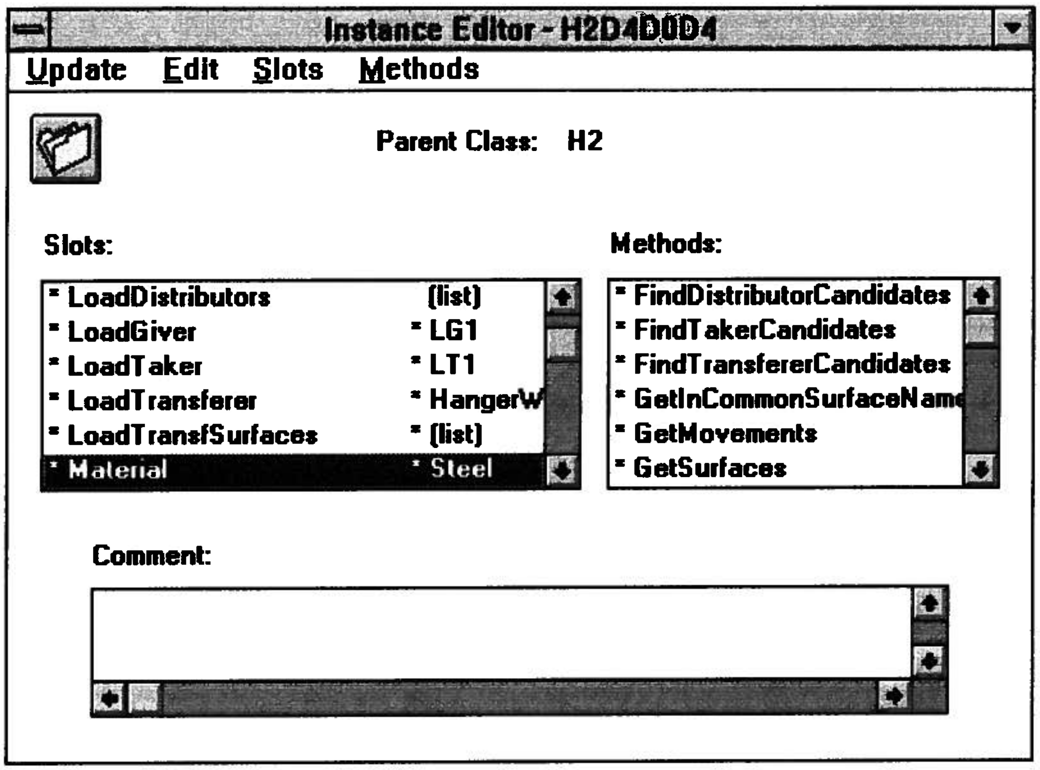


Figure 6.4 Inside a KAPPA-PC Object

case for an instance object named *H2D4D0D4*. The screen displays a list of object slots, their values, and a list of methods for the object. The slots and methods are wholly contained within the object, thus representing its description and behaviour that may be different from other objects. The asterisks beside a slot name, slot value, or method name, indicates a slot, value or method that has been inherited from another class object higher up in the hierarchy. Such without asterisks are local to the object, and if the object is a class object, may be passed along to siblings of the class object - a feature of inheritance.

By selecting a slot name *Material* in this example, the slot editor screen in Figure 6.5 is called up, allowing the developer to assign or alter slot values, and describe the nature of the information in the slot. The slot

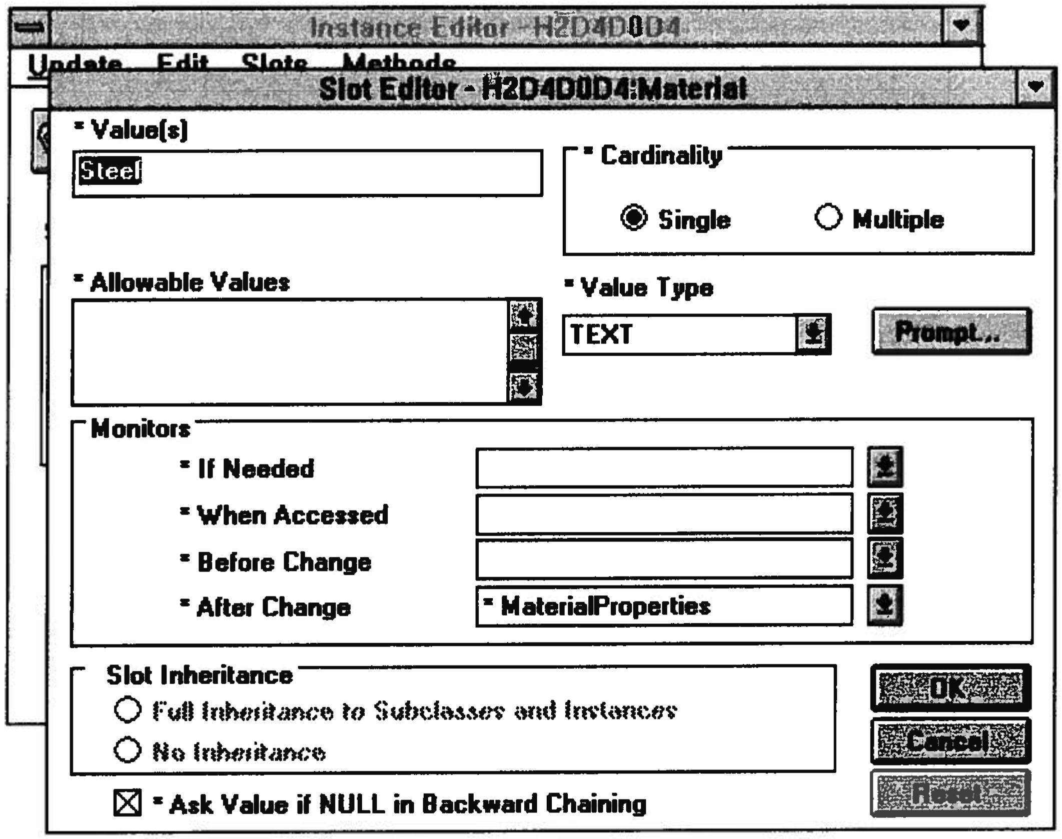


Figure 6.5 Object Slot Editor

also has trigger points called slot monitors that can initiate methods or functions depending on when a value enters the slot for storage. This feature allows for error trapping for entry data.

By selecting the method name *Material Properties* in the object screen of Figure 6.4, the method editor screen of Figure 6.6 is called allowing the developer to program object purposes (actions) using the C-like KAL language. KAPPA documentation describes a host of KAL functions that can be used to greatly simplify and speed development time, provided the user knows the function syntax and behaviour well. In this example, the user-defined function *GetMaterialProperties* is called with the input parameter Self meaning the name of the current object.

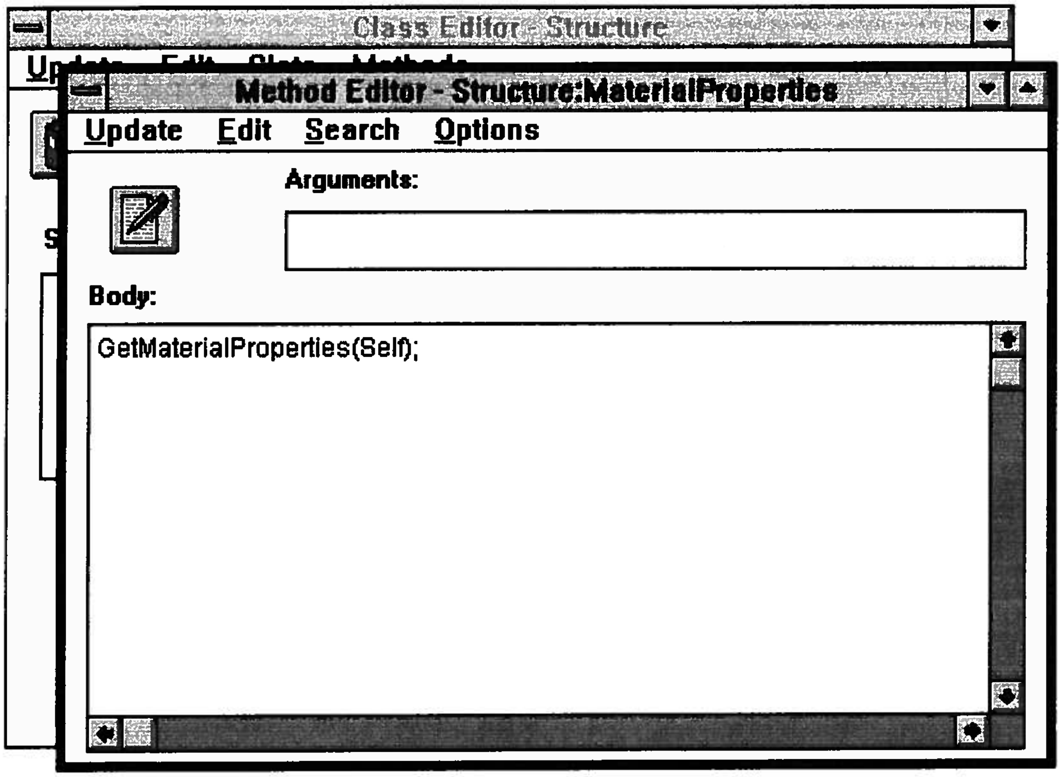


Figure 6.6 Object Method Editor

Functions are general procedures that can be used by any object and are held by the shell as programming elements external to the objects themselves. Access to the function editor is by means of a button in the Edit Tools box which calls up the screen of Figure 6.7, in this case, for the function *GetMaterialProperties*. This particular function initiates the forward chaining engine to process a set of rules for materiality.

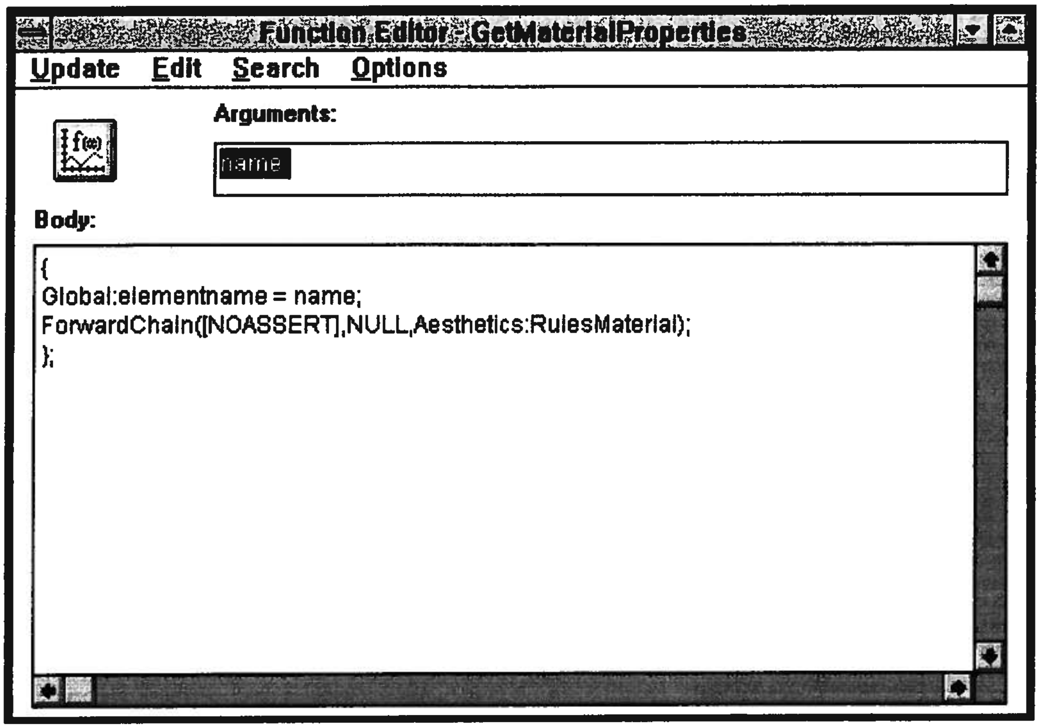


Figure 6.7 Function Editor

Similar to the function editor, the rule editor is also external to the objects and is accessed through the Edit Tools box. Choosing the appropriate button followed by identifying the rule name of interest, produces the screen of Figure 6.8, in this case for the rule *MATNaturalWood.* The text of this rule is written in an English-like IF-THEN format using slot names and test values as appropriate. The functions and rules interacting with attributes and behaviours of various objects is a fundamental idea of how the expert system shell representation basically works, which is very close to a real-world paradigm.

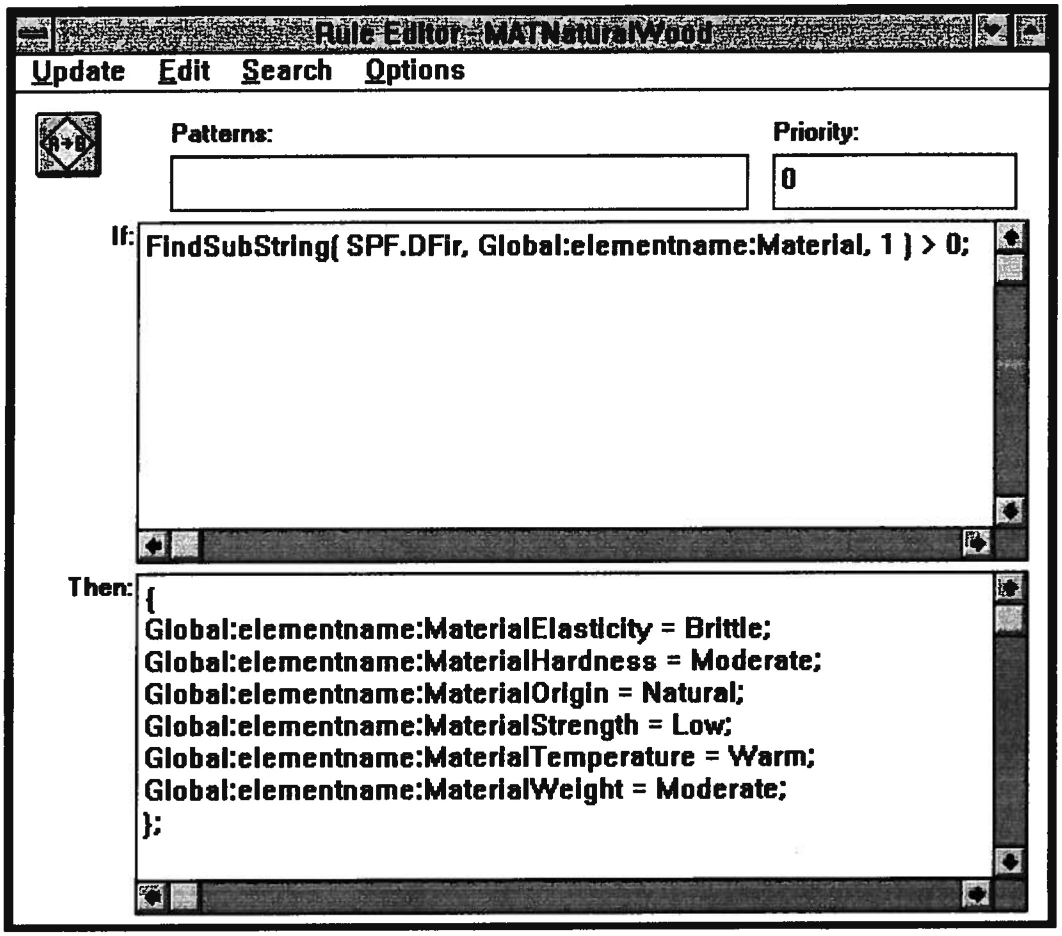


Figure 6.8 Rule Editor

The C-based expert system shell KAPPA-PC appeared well suited to achieve the goals of this research as a powerful development environment offering:

- rapid system development,
- reduced maintenance effort,
- true object-oriented environment,
- powerful rule-based reasoning,
- dynamic presentation graphics,
- intelligent links to popular software,

- open architecture, and
- standards adherence.

However, its limited graphics function palette and DDE iteration capability mentioned earlier precluded development of the completely envisaged system, and instead focused the research towards the implementation and synthesis of design knowledge.

6.2 KNOWLEDGE REPRESENTATION

This section outlines the implementation of knowledge representation and reasoning of timber connection design for the computer for the application of timber connection design.

6.2.1 System Architecture

A description of overall implementation architecture from a general viewpoint and discussion of each component in detail is offered. A global view of the system organization appears in Figure 6.9.

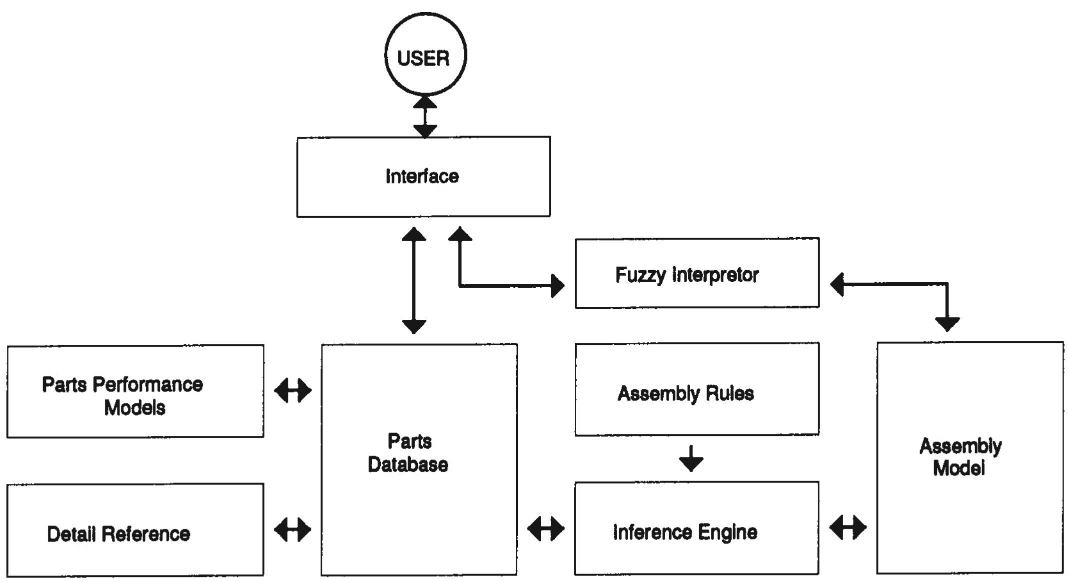


Figure 6.9 Global System Architecture

6.2.1.1 *Interface*

The interface is the communication port between the user and the computer tool. It is represented by the expert system shell by a series of screen objects containing image objects that transfer information between the program and the user. In some cases, the image objects perform translation or interpretation functions before data is communicated to the program, as in the case of a fuzzy interpreter. Ideally, the interface would be such that a life-like representation of objects could be displayed and easily manipulated by the user in real time while receiving corresponding computer responses according to the user's demands. Due to KAPPA's limited graphics command palette, only symbolic representations and linguistics were possible in order to interface with the user. This will be evident in the implementation Figures to follow.

6.2.1.2 *Fuzzy Interpreter*

The fuzzy interpreter translates numerical information to linguistic information or vice versa as required by the program by implementing the membership function concept as previously described. Each membership function is stored as an object with key value attributes and translator methods.

6.2.1.3 *Assembly Model*

The assembly model consists of a series of objects arranged in a tree-like fashion to represent the design world of connections. The assembly model is a composite of all the components meeting at the connection: members, hardware, and fasteners. The model can be divided into two parts: one which purely represents each of the physical objects of the *Structure*, and another which can represent the collection as a *Product*. The subtle difference is that the Structure contains physical information about each specific part, while the Product contains assessment information that is relative among the parts and about the entire assembly itself, such as total cost, total performance evaluation, satisfaction of global design requirements, etc. A *Structure* consists of *Members* and meeting points, *Joints*. Joints consist of *Hardware* which bridges the joint gap, and *Fasteners* which connect Hardware portions (surfaces) with Members. A *Product* can consist of various *Assembly Options*, which directly correspond to possible physical combinations of Hardware and Fasteners for each Joint. A more complete description of this implementation will be presented later in Section 6.2.2.

6.2.1.4 *Parts Database*

The parts and surfaces data bases for the various connection part objects are contained in a hierarchically arranged object tree in which object attributes are filled by the reading-in of external ASCII text data files. An extract of the text data files appears below in Figure 6.10. It is important to note the scant amount of information required for the system for an object, a beneficial feature that greatly reduces the burden of adding new objects to the system. It is envisaged that the data addition task could also be automated in a manner that simply queries the user through a window interface for information about the object that is to be added.

object database

PartsData HeavyTimber

Part	Description	Class	Contact	Material	Cost	Availibility	EaseofUse	Visbility	SurfaceClass	Fire
Ct	ColumnContinuousAtSupport	Columns	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
C2	ColumnEndingAtSupport	Columns	End	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
R1	RafterContinuousAtSupport	Rafters	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
R2	RafterEndingAtSupport	Rafters	End	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
R3	SplicedArch	Rafters	End	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
R4	SpacedArch	Rafters	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
P1	PurlinContinuousAtSupport	Purlins	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
P2	PurlinEndingAtSupport	Purlins	End	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
B1	BeamContinuousAtSupport	Beams	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
B3	BeamSplicedAtSupport	Beams	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
B2	BeamEndingAtSupport	Beams	End	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown

surface database

SurfaceData HeavyTimberParts

PartClass	Name	Owner	LocationPlane	MovementFastened	MovementUnfastened	MatingSurfaces	PossibleDistributors
SurfAngles	a	Taker	XY	xyzXYZ	z	bc	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfAngles	b	Taker	XY	xyzXYZ	z	ad	None
SurfAngles	c	Giver	YZ	xyzXYZ	x	ad	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfAngles	d	Giver	YZ	xyzXYZ	x	bc	None
SurfAnglesSP	ab	Taker	XY	xyzXYZR	z	cd	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfAnglesSP	cd	Giver	YZ	xyzXYZR	x	ab	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfAnglesSS	ab	Taker	XY	xyzXYZ	z	cd	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfAnglesSS	cd	Giver	YZ	xyzXYZ	x	ab	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone

Figure 6.10 Object and Surface Text File Database

In the tabular object database file, the information for each connection part is contained on one line. The Part column contains a unique user-specified part identifying code followed under the description column by a unique part name. The part name is also used to name the part object (instance) in the system object tree which is the reason for the uniqueness requirement. The Class column contains the name of the pre-defined system object class to which the part instance belongs. The Contact column contains the Giver/Taker contact code, i.e. SideSide for the object. This code represents the possible contacting surfaces that this particular object might have with the Giver and Taker. The Material column identifies the object material by name, i.e. Steel, or D.Fir. The names are predefined by the system and are wide ranging. The Cost column contains the cost factor for the object. The

Availability and EaseOfUse columns contain short standard descriptions on the availability and ease of use of each part. The Visibility column records the allowable visual possibilities that an object may take on in any design. The SurfaceClass column records the name of the system object class where object surface information is stored (the link with the surface database). Finally, the Fire column records the allowable fire protective possibilities that an object may take on in any design, i.e. Buried, or Exposed, or BuriedExposed meaning either one.

In the surface database file, which also is tabular in nature, the information for the surfaces of each connection part is contained in successive lines, one surface per line. This seen in the SurfaceAngles part class of Figure 6.10 which has four surfaces. This is the same Angle object shown in Figure 2.3. The PartClass column identifies the surface class corresponding to the object instance, and the Name column contains the surface instance name of the corresponding surface class. For example, surface *a* is a surface of *SurfAngles* as is seen in Figure 2.3. The Owner (perhaps join-owner) of this surface is identified in the next column as Giver or Taker, and the LocationPlane identifies the plane in which the surface is located according to the orientation of Figure 2.3. The MovementFastened column contains the movement code for the surface consisting of a list of axis directions in which the surface is restricted from movement if the surface is penetrated by a distributor. For example, xyzYZ means that the surface is free to move in the X and R directions of Figure 2.3, but not in any other directions (xyzYZ). The MovementUnfastened column contains the movement code for the surface consisting of a list of axis directions in which the surface is restricted from movement if the surface is *not* penetrated by a distributor. Basically, the longer the string length of the movement code is, the more fixity the surface has. The MatingSurfaces column contains the Name(s) of contacting object surfaces. Finally, the PossibleDistributors column lists in a string the possible distributors that a particular object surface may transfer load to. This is the vitally important load path connection between the Transferor and the Distributor.

6.2.1.5 *Parts Performance Models*

The parts performance models consist of the Transferor analysis engines that can be high-level language programs that remain external to the main program. They are identified to the main program through a Transferor class attribute. By calling the attribute, a program of the name of the value of the attribute is called for execution. These specialized programs are characterized by their algorithmic, arithmetic, knowledge nature well-suited to high level language programming using *C*, or spreadsheets. Each program should be stand-alone; a feature which makes future modification, expansion, or elaboration, easy; and draw on object attribute data (slot values) for input/output.

6.2.1.6 *Detail Reference*

As a workaround to the limited graphics ability of the expert system shell and a feature that would not be necessary in a complete parametric graphics object solution, the detail reference is a Transferor attribute which can store the name of a bitmap file of a general image of the Transferor detail in question. This general image can be displayed on demand. A group of these detail bitmap files can be stored in a detail reference directory. However, the work to generate the bitmaps, and the storage volume the files imply requires serious reconsideration of this approach. It is a simple attempt to supersede the shell's severe generative graphics limitations, which in reality prescribes the use of a much better graphics-based representation program approach. However, according to KAPPA literature, CAD linkages are easily established through object attributes - a feature that was not explored in this work.

6.2.1.7 *Inference Engine*

The inference engine is proprietary to, and wholly contained within, the KAPPA system kernel. Both backward-chaining and forward-chaining engines are present which interact easily using object attributes as variables and an IF-THEN rule structure. The reasoning behaviour of these engines are well represented with examples in the KAPPA documentation.

6.2.1.8 *Assembly Rules*

Rules are easily created using a rule editor within KAPPA in a very common English-like text structure. The main objective of the rule is to test object slot values (attributes) and perform some activity if the rule tests positive. Groups of IF-THEN rules are stored in rule lists which form agendas to be acted upon by the inference engine. Rule lists reflect rule groupings of interest to a particular concern such as: fire, visibility, movement, etc. The rules test object attributes (slot values) and either set new or other object attributes, or call some function that does. Rules are deemed to have been fired if the IF portion matches current value(s) of an object attribute(s). Functions and interface object methods are used to initiate procedures or the rule chaining engines as appropriate to order the flow of the design process or perform critical evaluation of the evolved design, display results, or a number of other similar tasks.

6.2.2 Object Tree Description

The intention of this section is to describe the object implementation of this KAPPA application to design timber connections within the context provided by the previous Section. Here, a more detailed discussion of the joint object tree structure and slot categories of basic elements are presented. The information in the figures to follow deals with the design of a connection for a beam framing into a the side of a girder shown in Figure 6.11.

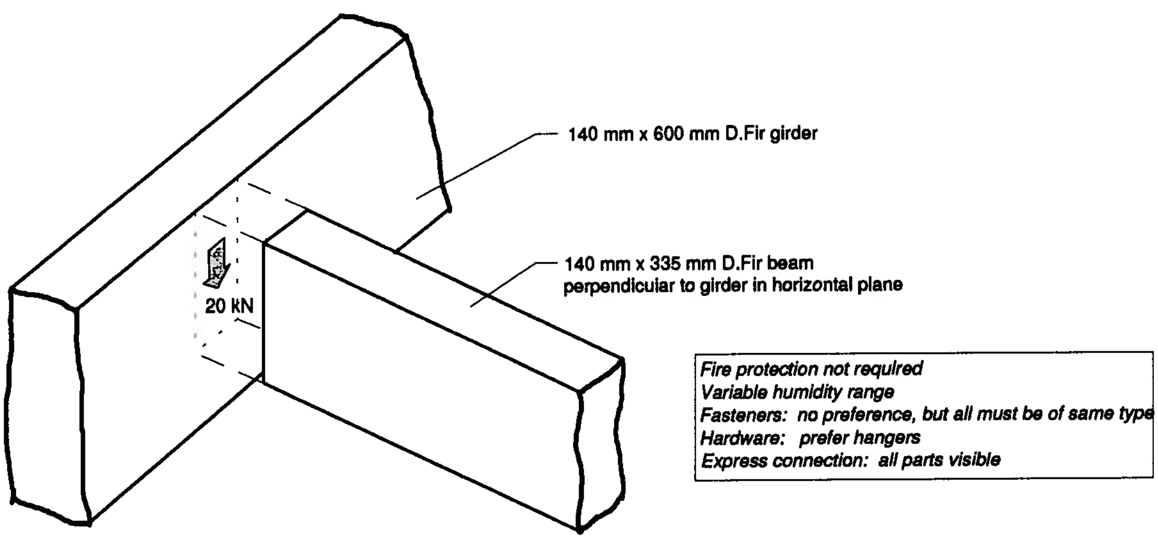


Figure 6.11 Illustrative Problem

Many of the system architecture components of the previous Section are implemented in the object tree of Figure 6.12. The unboxed Global instance object contains all temporary variables (slots) and constants used by the system or by any object. The remaining boxed objects indicate that a branching into more elaborate tree structures exist. The Menu class would contain many objects for various application windows - commands that work very similarly to common menu bars in the Windows environment. The Image class contains subclasses and user-

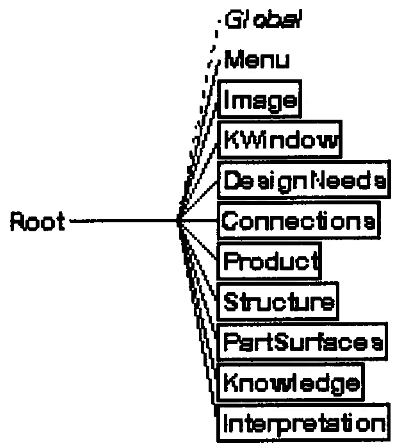
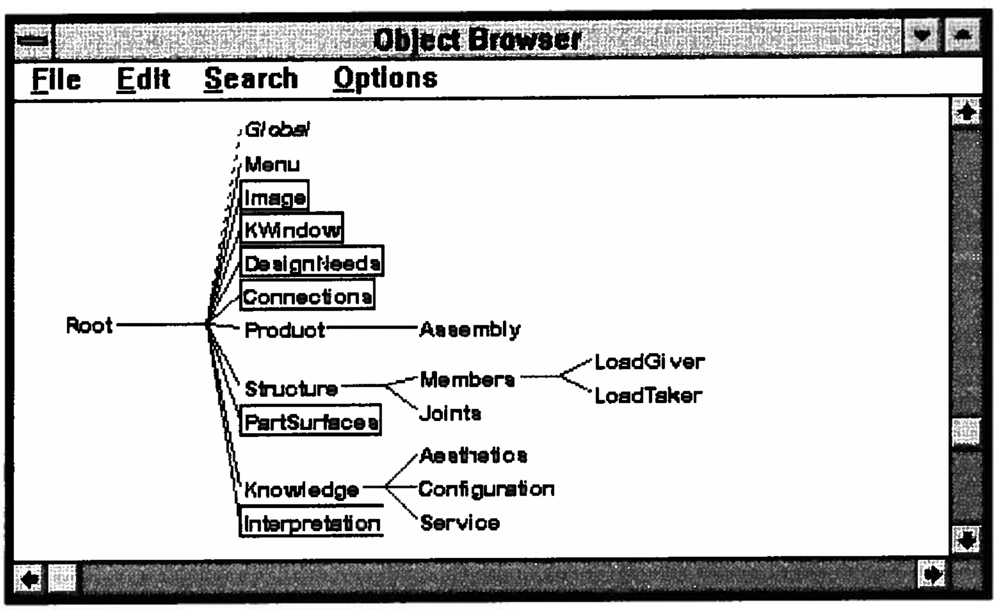


Figure 6.12 Implementation of System Architecture

defined instances of various window objects, such as buttons, sliders, graphs, etc. that may be used to build an application interface. The KWindow class contains instances of interface windows that can be constructed by the user. The SESSION subclass of the KWindow class, as well as the Image and Menu classes, and the Global instance are provided by the shell product and remain as constant entities of any application. Hence, the user interface is built with developer-defined instanced objects from the shell's KWindows, Image, and Menu classes and subclasses. Completing the user interface are the contents of the Interpretation object tree which contains membership function objects for a variety of linguistic descriptors, and the internal object methods that permit the objects to convert quantitative data to qualitative data and vice versa using the techniques provided in Chapter 3.

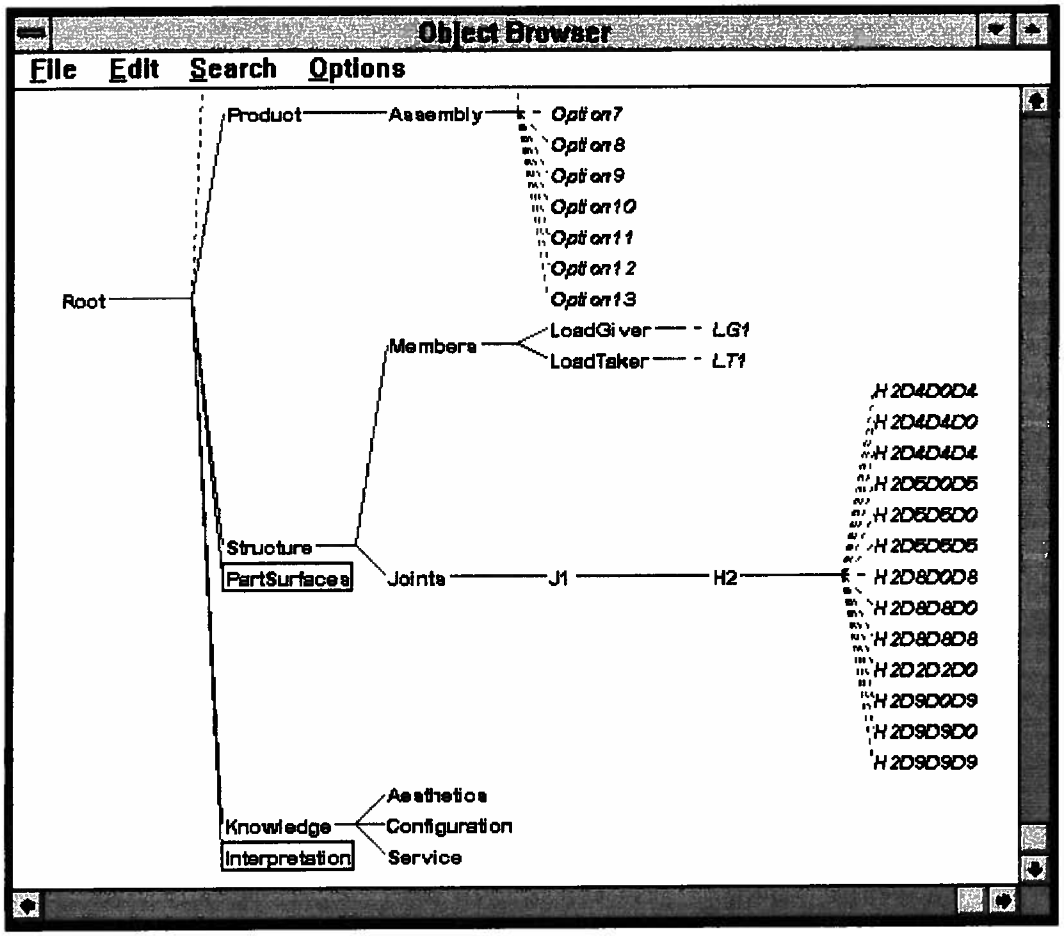
The main components of the developed application's object tree appearing in Figure 6.12 are the Knowledge base, the Structure tree, the Product tree, the Design Needs tree, the Part and Surface databases, and the interface/fuzzy interpreter components. Implementation of the Assembly Model is done in two parts; one to represent the evolving design as given in the Structure object tree, and another to represent acceptable design alternatives given by the Product object tree. These tree structures are shown in the Browser Window of Figure 6.13. The Structure tree is interesting in that it is dynamically constructed from the joint out. The joint object creates Transferor objects, which create instances that reflect the possible Transferor/surfaces/Distributors combinations for a connection. The dynamic creation is done with functions and rules implementing the design process as previously described. In addition to generating Transferors, Surfaces, and corresponding Distributors *on the fly* (dynamic class/instance creation), the joint tree handles multi-surfaced (>2 surfaces) Transferors.

A closer look at the Structure tree of Figure 6.14 reveals the coding of the various objects representing different combination possibilities of Transferor and Distributors for a connection design. In the Figure, the Transferor part code *H2*, representing a type of hanger, is carried forward in the naming of the transferor/distributor combination instance - a different instance for each Distributor combination that can be allowed for each of the Transferor surfaces. Since this particular hanger has three surfaces, and a Design Need is that all the fasteners must be of the same type in this example, appropriate distributor selections are made for each of the Transferor surfaces, Distributor part code for each surface. Hence *H2D2D2D2* would indicate a Transferor of Type *H2* with Distributors of Type *D2* penetrating each of the Transferor's three surfaces. The class relationships of Figure 6.14 depart slightly from the standard accepted consistent *part-of* and *is-a* constructions as commonly found in the literature. The reason for choosing this combined approach was to realize a significant savings on programming overhead by using the shell's inheritance feature.



start

Figure 6.13 Structure and Product Object Tree Representation



end

Figure 6.13 *continued*

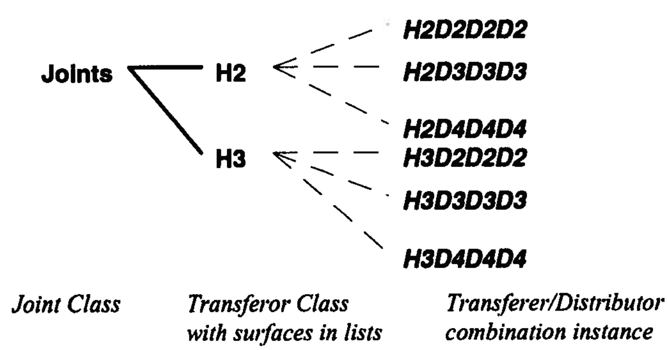


Figure 6.14 Dynamically Generated Joint Object Tree

The Transferor part code Combination instances are *critiqued* by the rule network to create design Options which are stored as part of the Product class. Instances of this class contain assembly cost/benefit information and pointers to assembly objects for variety of connection designs, one design per Option. Design Options are then ranked by an interface image object according to cost/performance as stated previously and presented to the user.

Part and surface information stored externally as ASCII text files, is represented by the tree structure of Figure 6.15. Of immediate note is the tree's natural hierarchy which represents a kit of parts for heavy timber connection design. Also, it is possible to extend this tree to encompass other forms of wood design, such as light

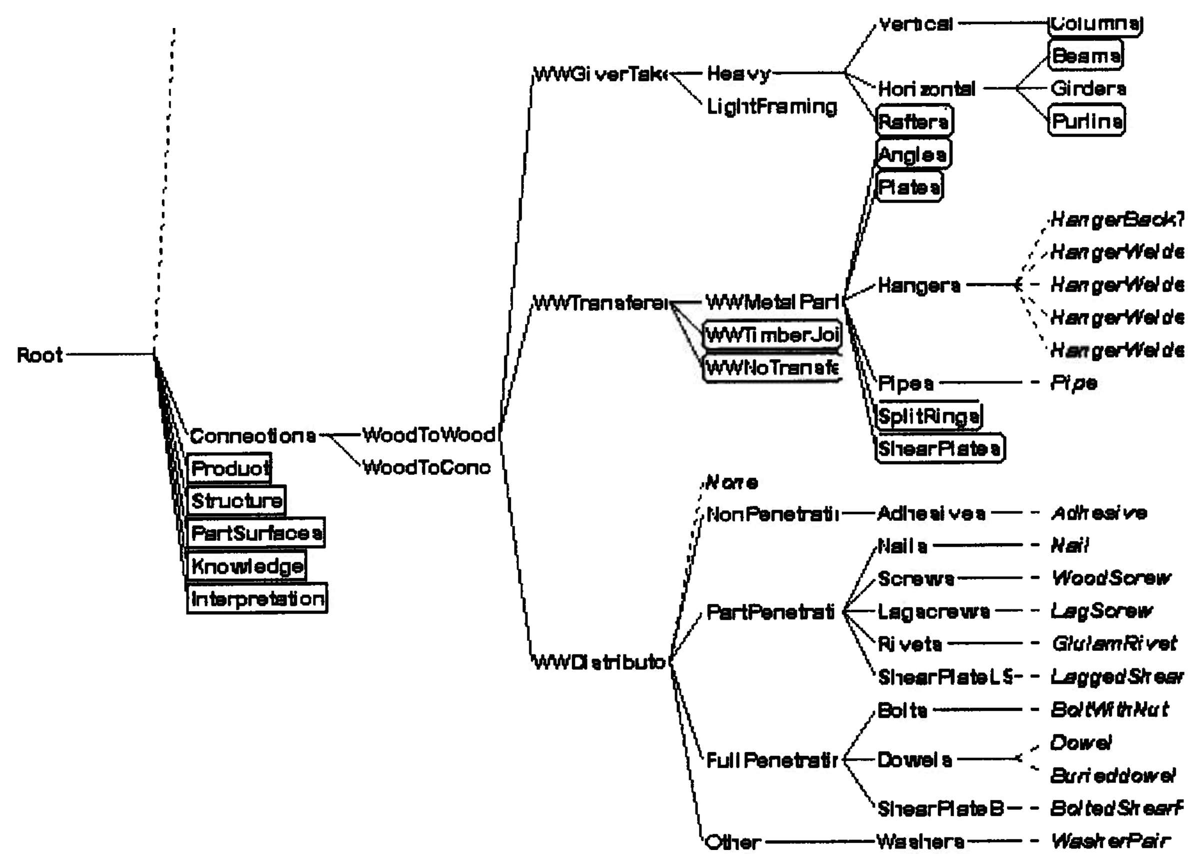


Figure 6.15 Parts Object Tree

framing, or even other materials, such as steel. The important point here, is that an instance of this tree contains within its slot values all the possible information, qualitative and quantitative, that any part can take on, much like a real world object. A pick from the parts kit, literally invokes all the design possibilities that this choice represents. The tree also acts as the bridge to an external CAD system such that CAD object properties can be captured, transferred, and stored as object slot values that can be used for design reasoning within the system shell. Similarly, the parts tree objects can contain external file names of typical bit-map representations for graphical display, or methods that will call external routines to calculate part capacities as featured in the study of Chapter 5. The Parts instances can be flushed and refreshed with data from the ASCII files at any time by a simple call to a file read function, hence updating of the system with new user-defined data can be done easily at any time. This is beneficial form of separation of data from the system control structure in that the knowledge base and control are independent.

An interesting feature of the parts object tree occurs in two objects: the NoTransferor, and NoDistributor. Each of these objects has particular qualitative and quantitative properties and attributes in the same manner as all other objects in the parts tree. These invisible objects account for the cases where members may be joined by: hardware and no distributors; or no hardware, just distributors, or nothing at all. The inclusion of these two objects allows the critical evaluation of connection designs where these objects exist, and also permits screening designs and suggesting remedies where fastener or hardware redundancy amounts to over-design.

The rationalization structure is contained in the Design Needs object tree and the Knowledge object tree shown in Figure 6.16. The Design Needs objects contain design constraint information that helps guide the reasoning process, such as appearance, group patterns, preferred fastener types, etc.. In some cases, slot values in image objects supplement the Design Needs criteria. This information can be thought of as goals that the user wishes the design to consider in generating solutions. The information can be scaled to individual elements such as fasteners, members, or hardware; or to groups of elements so that patterns can be considered. The Knowledge object tree consists of appropriate classed objects which can be likened to reference books. Slot values in the Knowledge objects are also appropriately named and contain lists of rule names that can be placed on the agenda

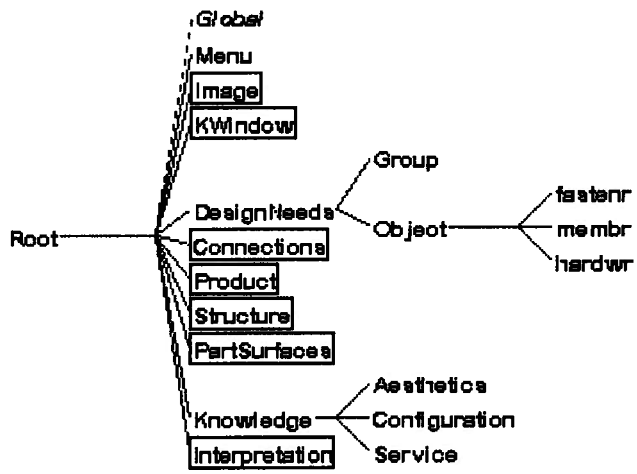


Figure 6.16 Design Control Object Tree

of the shell's reasoning engines for consideration. Thus, the slot values group the knowledge contained in rules into logical segments. As an example, each of the flowcharts of Chapter 2 can be represented by a slot name of type *list* containing the names of the IF-THEN rules for each branch in the flowchart. Some of the rule groups that exist in this application are rules for material description, fire resistance, movement, visibility, and in-service environmental performance.

6.3 RELATIONSHIPS AND REASONING

This section describes the implementation of reasoning as presented in Chapters 2, 3 and 4 and does so by means of the beam-to-girder connection example of Figure 6.11. Reasoning in this application takes place in three ways.

6.3.1 Object Methods and Functions

Some methods and functions are used inside the object itself to make it smart by allowing the object to respond appropriately to external messages sent to it. The messages normally consist of new slot value assignments that occur on demand external to the object. This type of reasoning is most often found in the interface image objects.

6.3.2 Inter-Object Methods and Functions

Some methods and functions are used to automatically relate object behaviour within a group. This polymorphic feature is useful in that if a single message is sent to three objects, for example; the three objects, through their inherent methods, may respond to the message in three entirely different ways. This feature is can be used effectively as a criticizing tool to investigate object "what-if" scenarios if a single change is made globally to the object collection.

6.3.3 Rules

Decision trees of rules (direct and fuzzy) used to constrain the design space are particularly effective when the data is linguistically based, as it is at times in describing the architectural world and dealing with issues of logic or meaning. The shell's handling of linguistically based data, particularly in the context of rules, was one of its strong points.

6.3.4 Implementation Example

To appreciate the reasoning issues involved, the Beam-to-Girder connection of Figure 6.11 will be configured using the developed system. The user interacts with the system through a number of windows that are logically categorized according to design information subject matter. When the system starts, the user sees the main window of Figure 6.17. The window contains a number of interface image objects: buttons and display-text boxes. Each of the buttons has a corresponding method which invokes a function when pushed. The system works on the premise that for every joint identity, there is one load giver, and one load taker. Thus a connection of multiple members consists of a number of joints within the connection. The LoadGiver and LoadTaker text boxes are crude symbolic representations of timber members, while the JointPossibilities box is a similarly crude symbol of connectivity. The member text boxes display detailed information about the member in particular, while the connection box lists possible design options that the user may choose to investigate further by selecting one of the option names. The text box at the bottom of the window informs the user of the design system status and progress at any point in the process. The initial system object structure is that of the *start* screen of Figure 6.13.

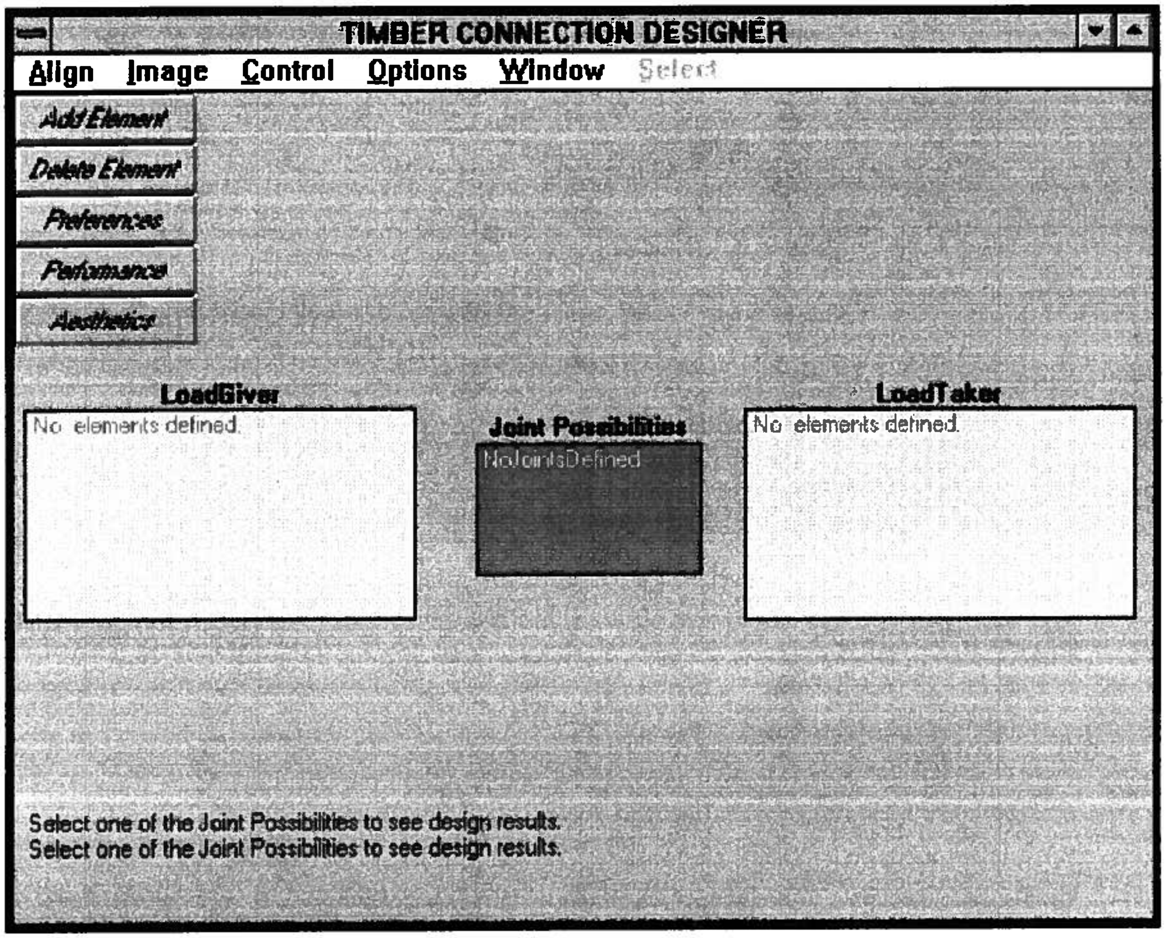


Figure 6.17 Initial System Window

To begin the design process, information about the elements to be connected must be provided to the system. By selecting the *AddElement* button, the window of Figure 6.18 appears allowing entry of identification,

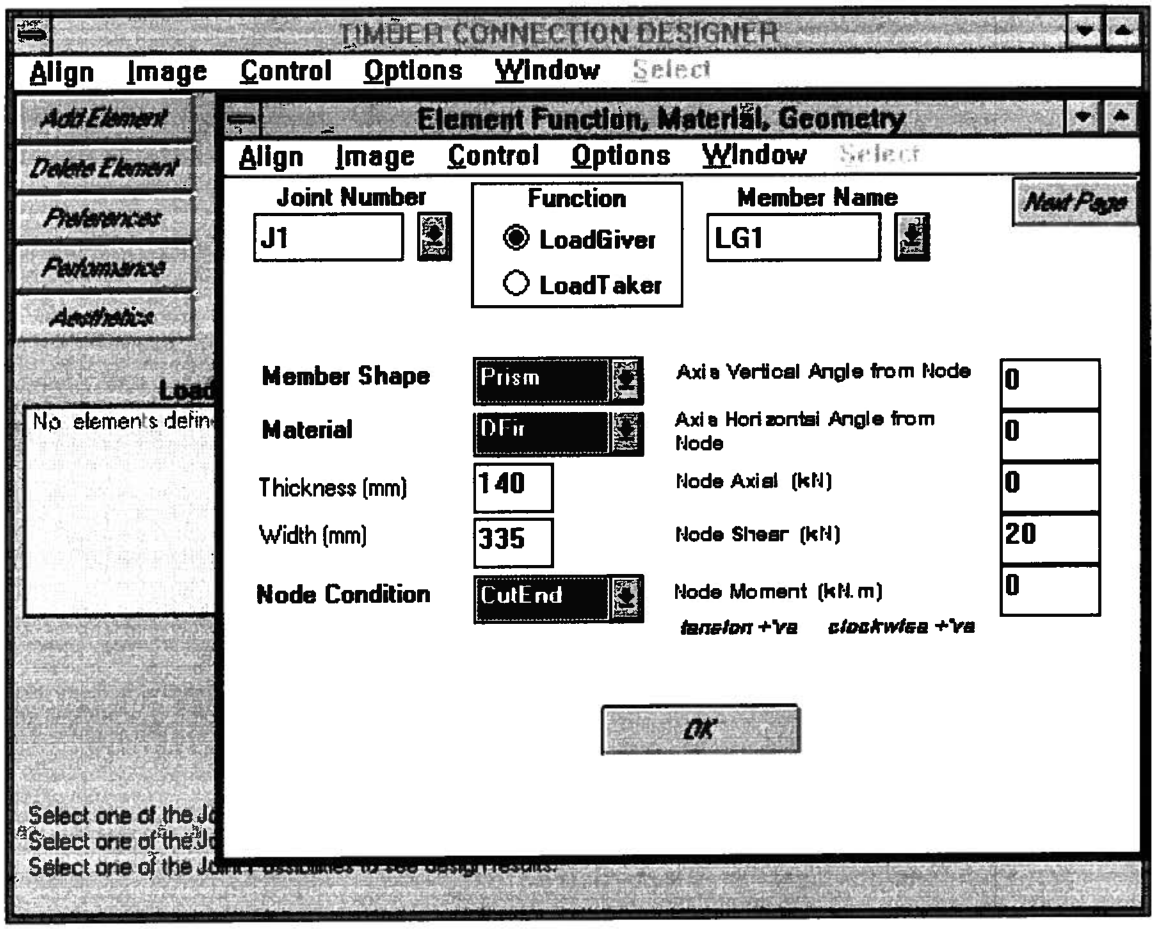


Figure 6.18 Element Quantitative Information Window

material, geometry and loading information about the element. In this Figure, information about the Load Giver is first entered. By choosing the *NextPage* button, Figure 6.19 appears allowing entry of element qualitative information. Most of the image objects in this window are drop-down boxes from which the user makes a selection

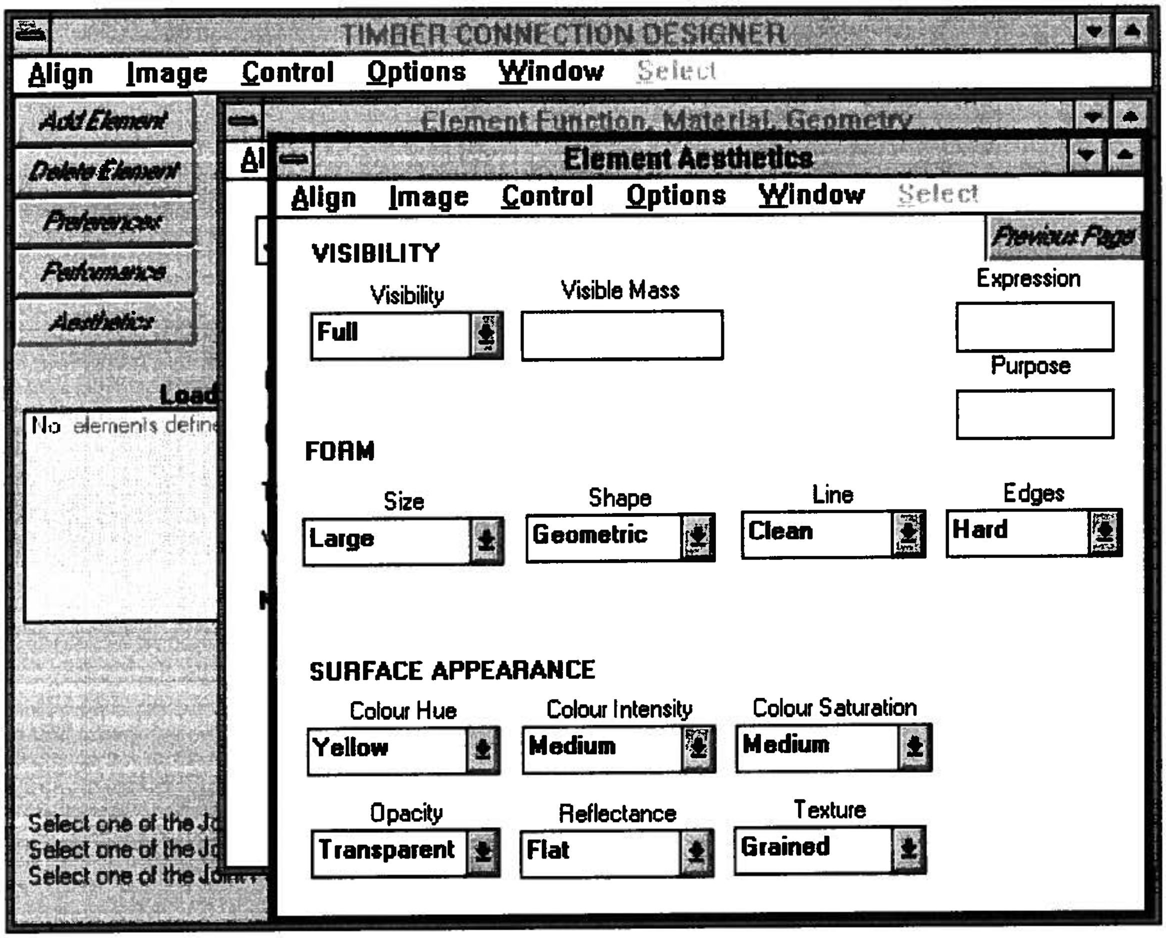


Figure 6.19 Element Qualitative Information Window

from a list. When the value is selected, the selected value enters the drop-down box image's Value slot, which calls the image object's method. The image object method calls a function that provides quantitative translation of the quality by sending a message to the interpretation objects. In this manner, linguistic/numerical translation occurs providing the bridge between the qualitative and qualitative worlds. This is significant, as often architectural reasoning is done in linguistic terms, even though the objects are numerically described as is usually the case in CAD systems and data structures.

By choosing the *PreviousPage* button, the user returns to the window of Figure 6.18. By selecting the *OK* button of Figure 6.18, the system creates the LoadGiver element in this case within the system object structure as a

LoadGiver instance object *LG1*. All of the entered properties are deposited as slot values with the instance's slot structure inherited from the member sub-class. Figure 6.20 shows the resulting main system window acknowledging the LoadGiver creation. If interest here is that the system describes this element as a Beam even though no such information was entered by the user. This information was inferred from the element orientation in space and applied loading type using a system of simple rules based on structural element definitions. Thus the created element classes itself within the context of the parts kit object tree.

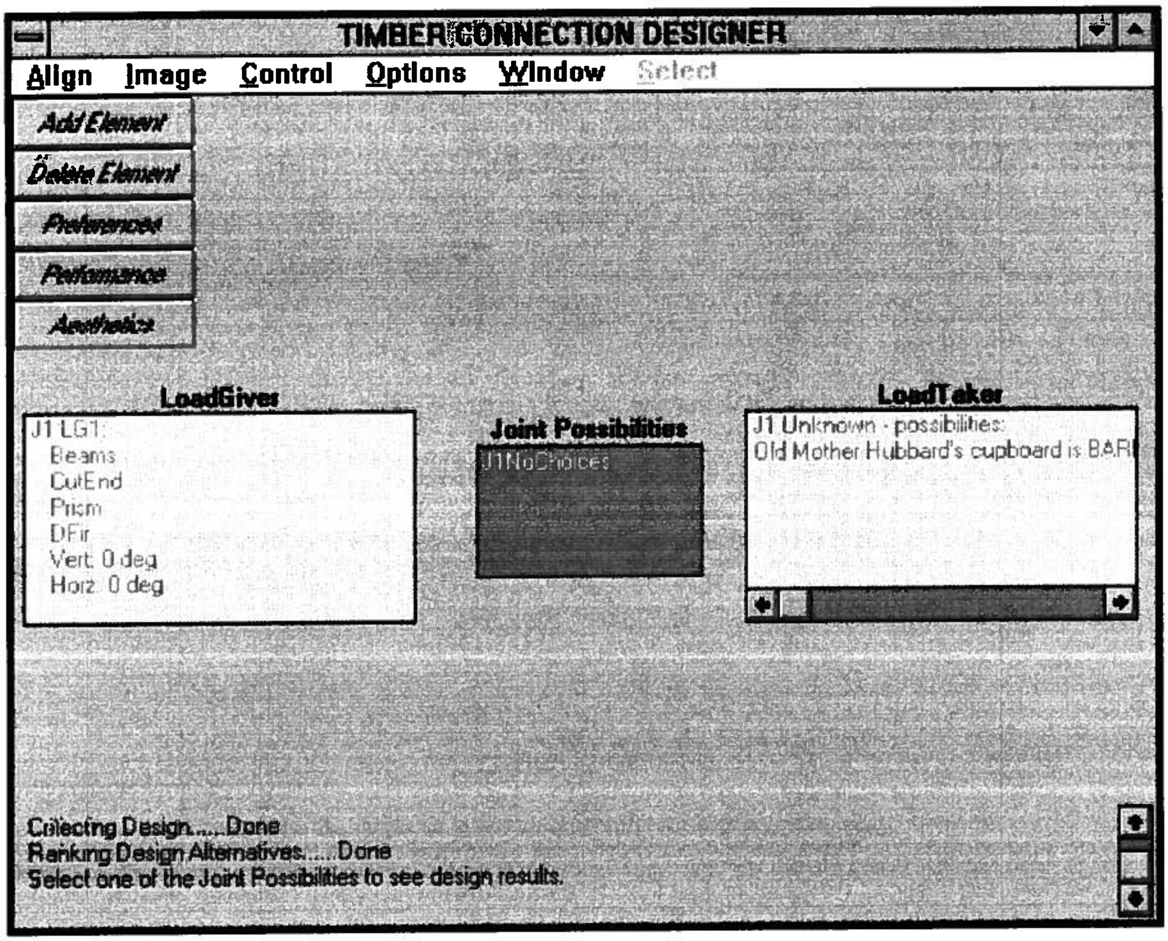


Figure 6.20 Load Giver Element in Main Window

The same procedure is repeated for definition of a Load Taker element as is show in Figure 6.21. If the element is connected to a joint where another opposing function element exists, then equilibrium is automatically checked and passing force values are entered automatically in the applied loading boxes. Choosing the *OK* button of this window , the system creates the LoadTaker instance object *LT1,* and since joint *J1* has been defined by the definition of the giver and taker, the process of designing an appropriate connection begins.

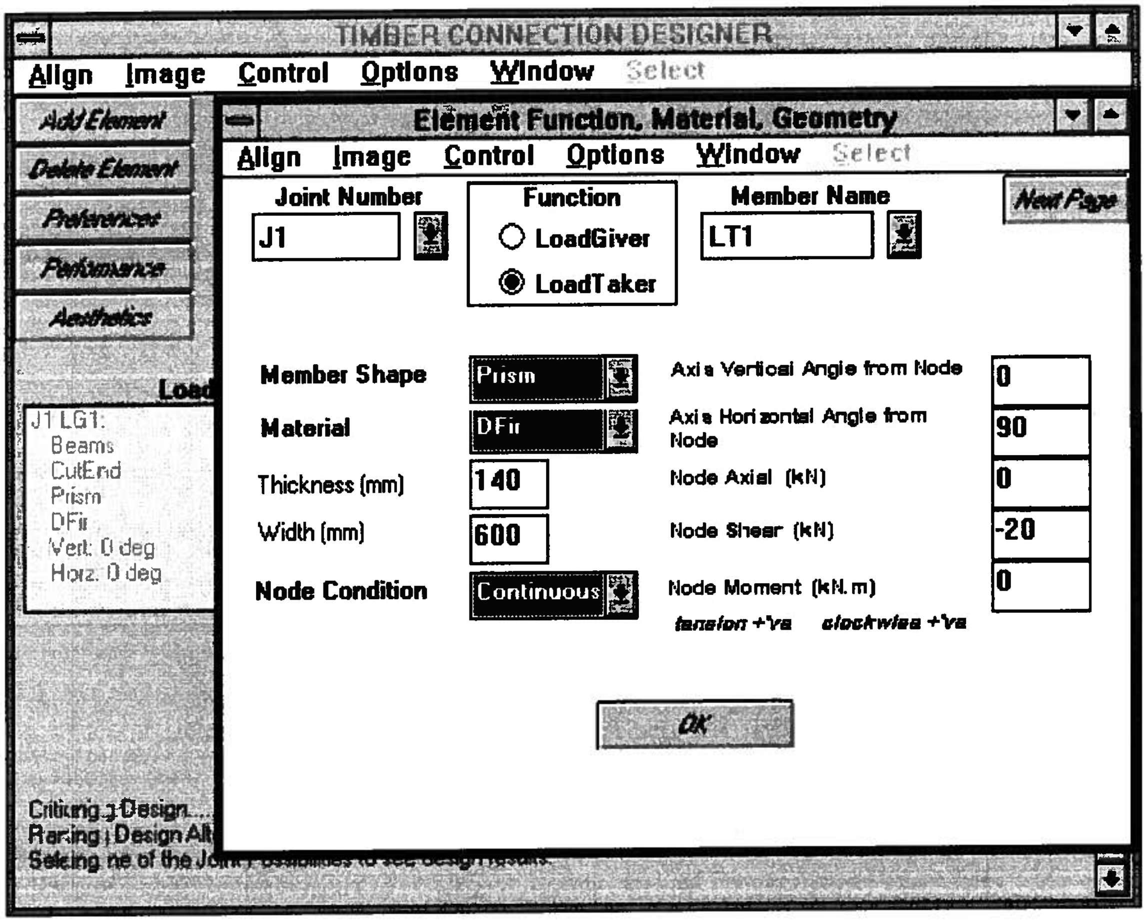


Figure 6.21 Load Taker Element Information

The *Performance* button of the main window allows the user to specify in-service performance criteria of the design in the window of Figure 6.22. The design requirements here provide design constraint data for fire performance, and environmental conditions. The unit cost input box allows the user to update the parts cost data at any time. The entry is a multiplier for the relative cost values stored in the data base. While actual costs for all objects as a group can vary, the relative costs among all individual objects is believed to be relatively static. The OK button closes the window and starts the design process.

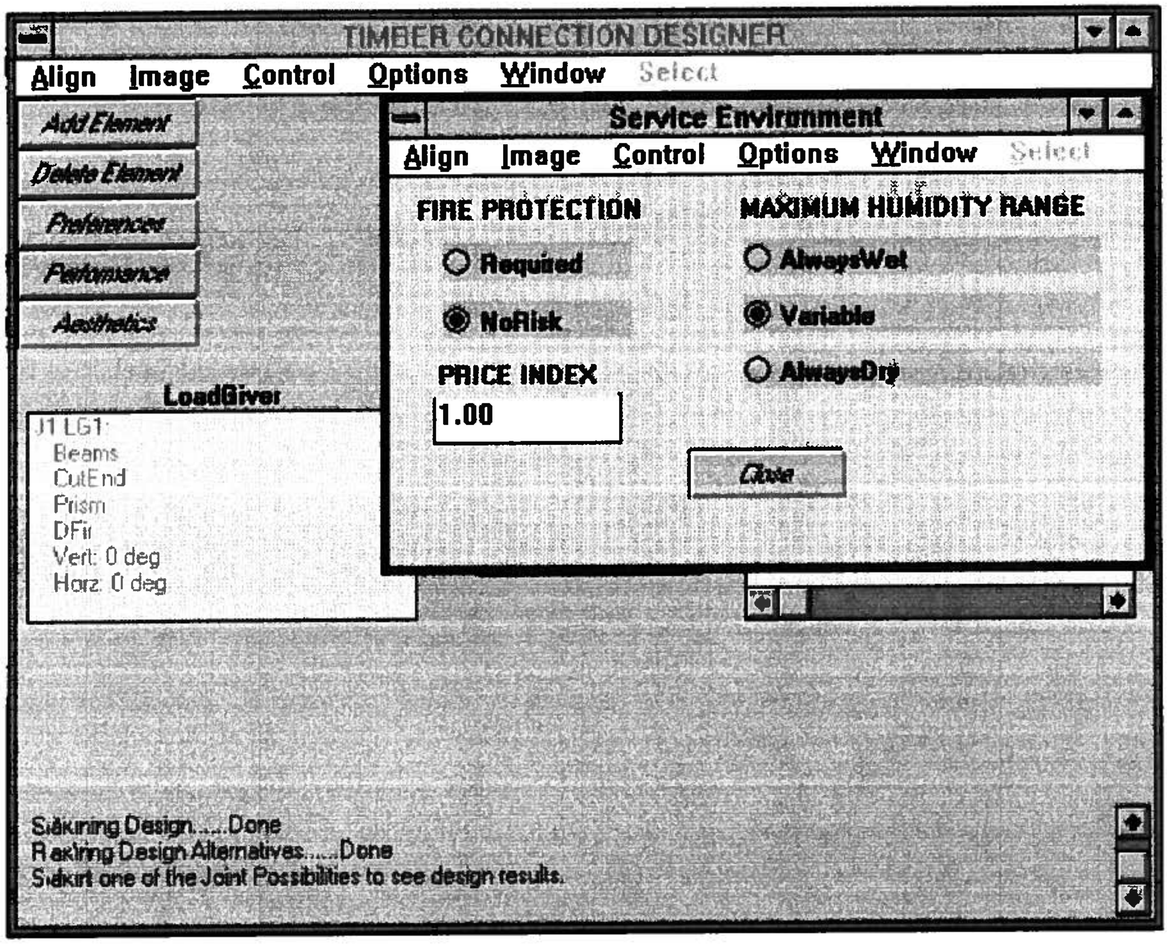


Figure 6.22 Service Environment Window

The Preferences button of the main window allows the user to specify aesthetic and preference criteria for the design in the window of Figure 6.23. These features also serve to constrain the design space in terms of grouping, appearance, and preferred parts. The appearance sliders work with the fuzzy logic interpretation objects to arrive at linguistic descriptors for the selected visibility in a range from totally hidden to totally exposed. For fasteners for example, the degree of exposure can range from complete burial, to countersunk heads, to flush heads, to raised heads, to fully exposed head and shank portion. The sliders can be altered by the designer to give the design some measure of visual expression from a clean, neat look, to an expression of brutality. The preference boxes allow the user to constrain the search of the parts data base to varying degrees of restriction from the complete parts kit or a selected variety to satisfy known availability criteria or some other reason. In this way, the

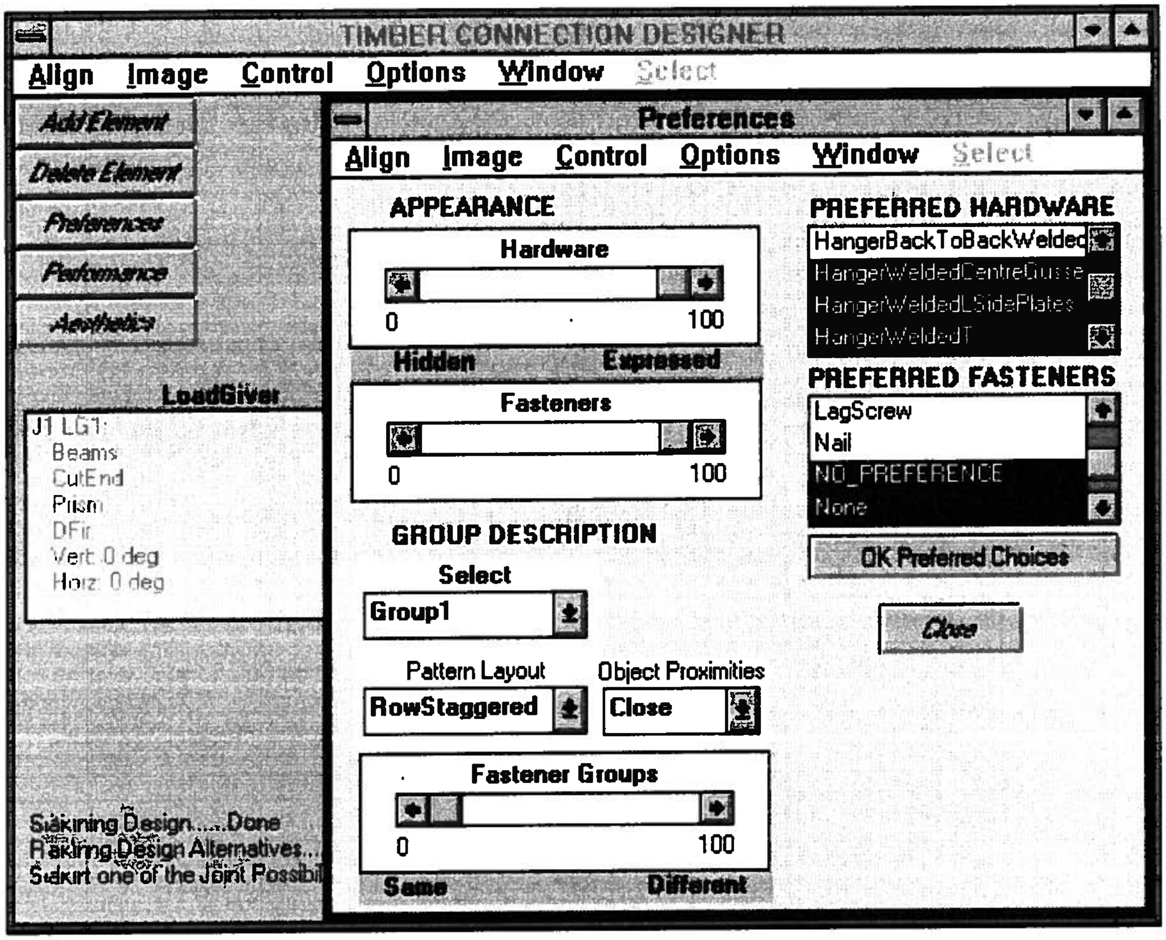


Figure 6.23 Design Preferences Window

user has some control over some of the design choices the system suggests. If the system doesn't offer any choices, the user can choose to broaden the search of the parts data by choosing more varieties. The group information is appearance information related to the fasteners on each transferor surface. For each surface connector group, the user can specify the pattern of the connector group layout, the proximity of the connectors within the group to each other, and whether the connector groups differ by type or are all the same. All of the information on this screen is stored in the DesignNeeds object slots as a constraint list of design requirements for design rules to check against. The *Close* button of the window initiates the design process and returns the user to the main system screen.

The design process begins with the system selecting a list of candidate transferors from the parts data base consistent with the stored design requirements and the rules for fire, compatibility, and visibility as presented in Chapter 2. For each member of the list of transferor candidates, a transferor object is generated in the Structure tree as a sub-class of its named joint (*J1* in this case) and named according to its part code. Each transferor object inherits all the attributes and slot values of the joint while adding a few more information pieces that relate to transferors, such as surfaces.

The next step, knowing the possible transferor objects, and their related surface information that is cross-referenced to the surfaces data base, the system selects for each transferor, distributor candidates for each of the transferor surfaces from the parts data base consistent with the stored design requirements and the rules for fire, compatibility, and visibility as presented in Chapter 2. For each transferor and combination of possible surface distributors, a connection object is generated in the Structure tree as an instance of its Transferor object and named according to the parts code for its transferor chained with the parts code of the distributors on each of the transferor surfaces.

Once the system finishes creating the connection instances, the entire rule base is brought to bear on the selection to critique and rate the selections based on design requirements for appearance, fire, in-service performance, and cost. The critiquing is done in a forward chaining mode and removes only totally incompatible connection instance objects from the Structure tree. More compatible connection objects remain for further consideration. The remaining connection objects are identified and referenced by corresponding Option instance objects created by the system as instances of the Assembly sub-class. The Option objects contain component references, total cost for the connection, and total demerit rating for the connection. A list of the connection instance object names is sent to the connection image object in the window of Figure 6.24. The image object sorts the name list according to best cost-benefit criteria as given in Chapter 2, and presents the ranked list to the user in the object display. By selecting any Option name in the object, the window of Figure 6.25 is shown providing all the pertinent details of this Option selection, such as: transferor selected, distributors selected for each of its surfaces, what kind of problems can be expected with this design, and a cost estimate for this option. Using bit-

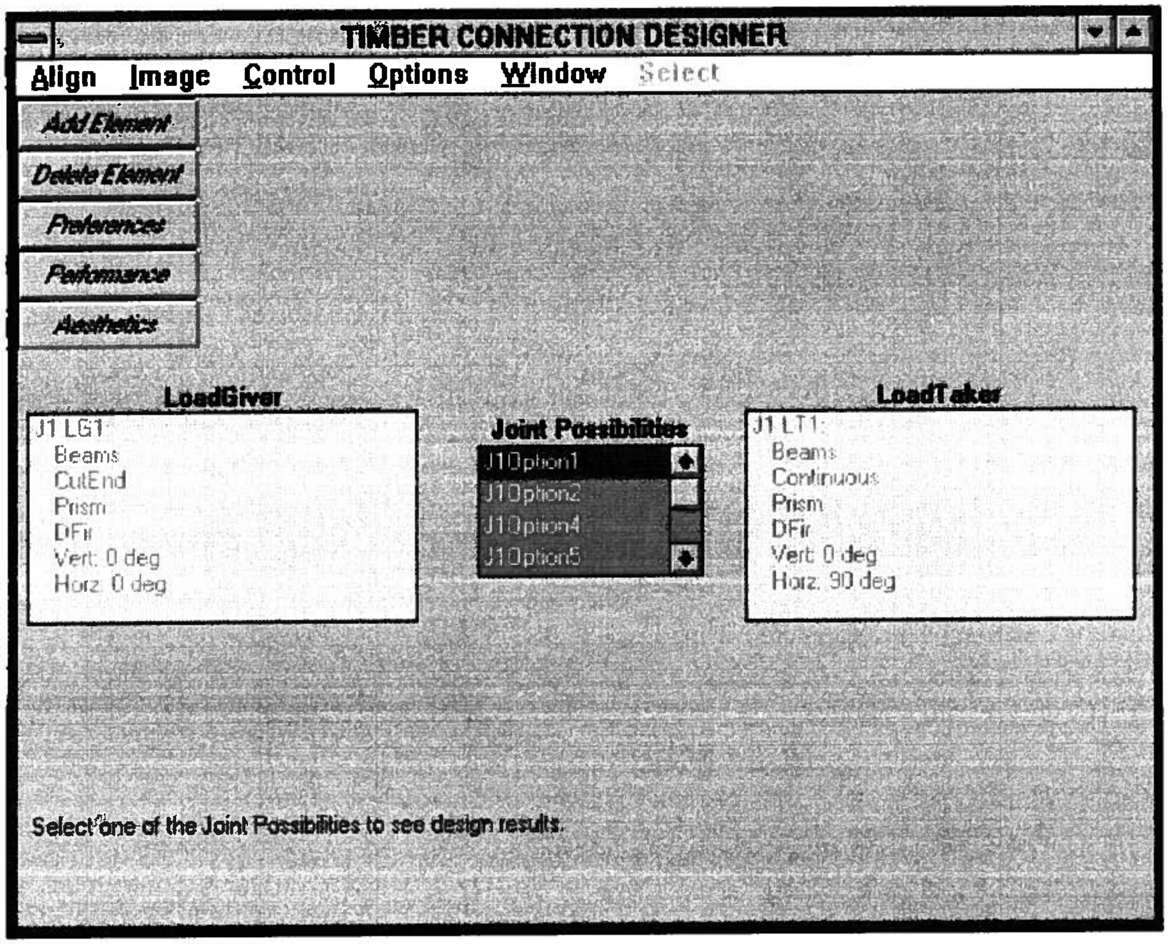
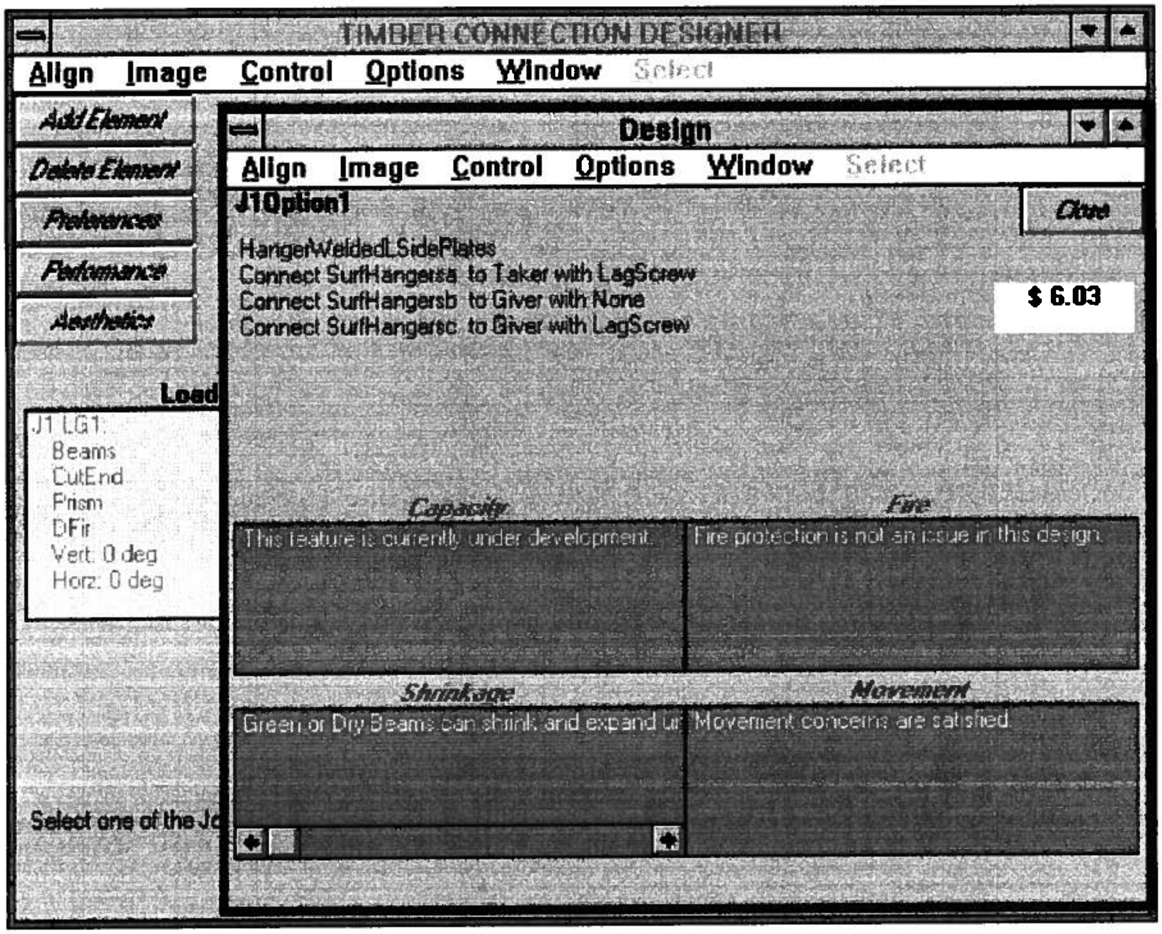


Figure 6.24 Design Complete Window

map image references from the parts data base proved to be too storage intensive as well as inflexible in showing installed variations. A future development of a direct CAD link here could provide the most pleasing and appropriate graphic solution as it also allows the user to directly engage the design while having the support of expert guidance and feedback. The development of this feature was considered not to be within the scope of this thesis. Figure 6.25 shows the resulting design for the "best" Option, while Figure 6.26 shows the resulting design for the most undesirable but still possible suggestion. The remaining design options fit somewhere in between. The final system object tree showing all the generated objects appears on the right side of Figure 6.13.



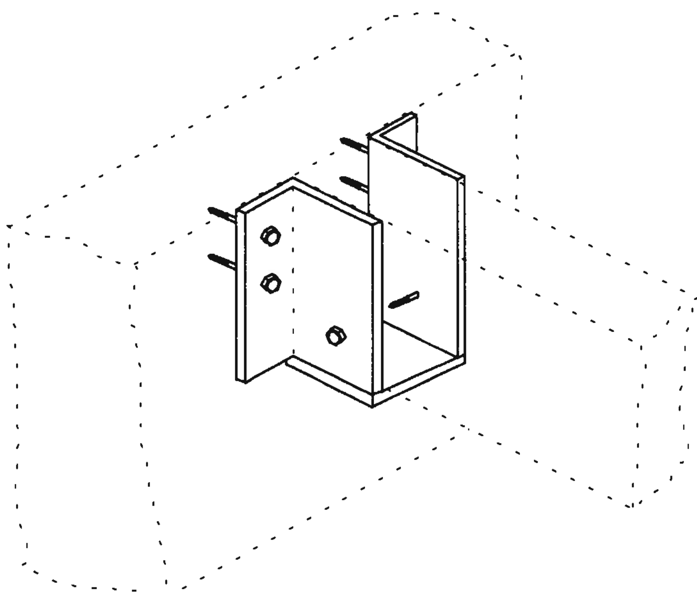


Figure 6.25 Best Design Choice

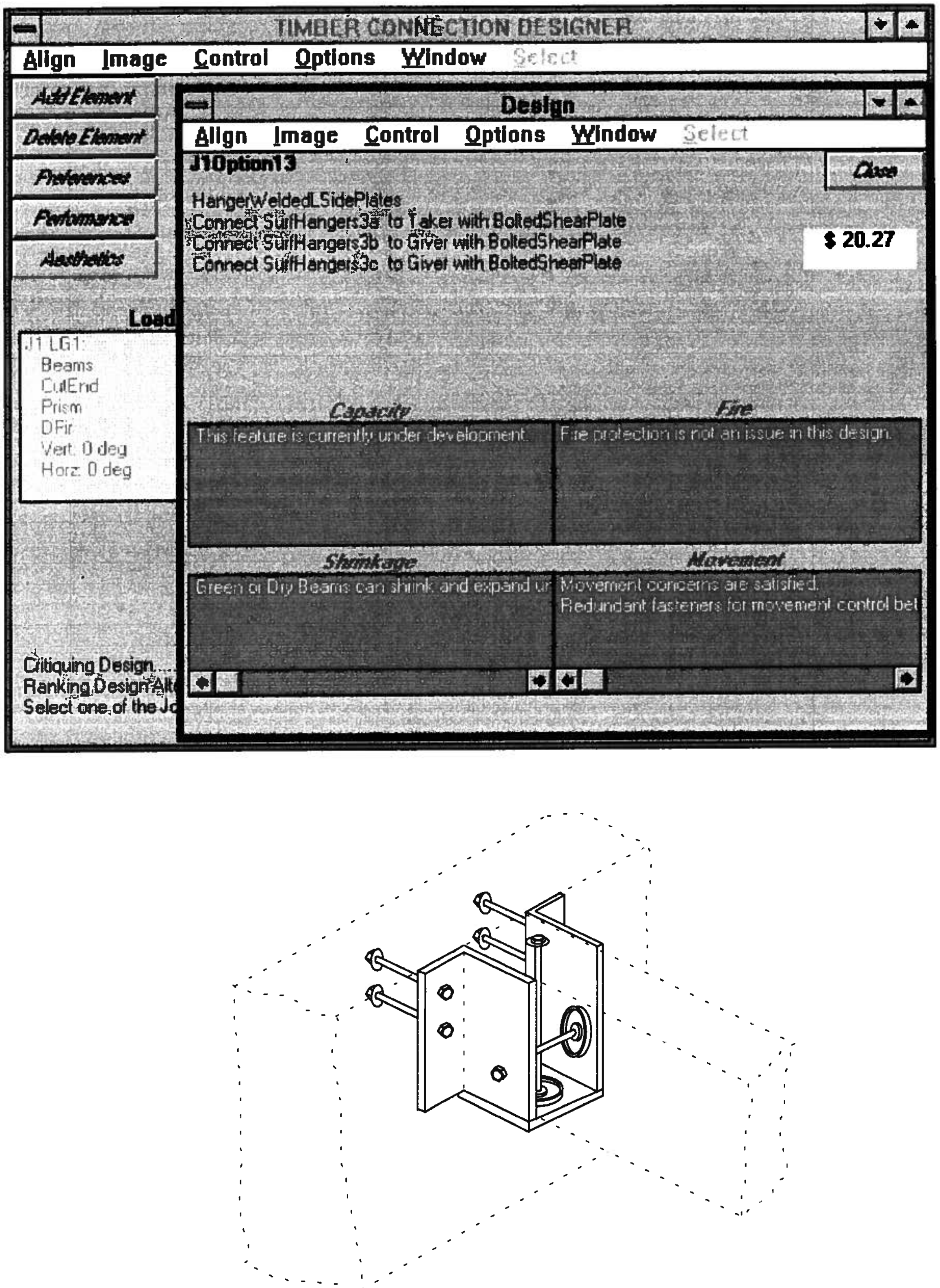


Figure 6.26 Poor design Choice

From here, it is up to the user to weigh suitability of the design considering the information provided. If none of the suggestions are appropriate, then the user, armed with the new knowledge of what has been tried, needs to alter the design requirements and repeat the process from any of the information windows. The problem structure stays inside the system until the user deletes the elements from the system using the *DeleteElement* button in the main window. Another window is available to the user by choosing the *Aesthetics* button. Figure 6.27 shows the Global Aesthetics window which gives the user direct access to membership definitions contained by the *Interpretation* objects that are used in translating linguistic/numerical information. This provision is important in that it allows the user direct control over what the user perceives the meaning of the linguistic terms to be. Some of the membership functions stored in the present systems are the result of the findings from the studies in Chapter 3, which are averaged from a population, thus reflecting an average interpretation. The system uses these constructions in interpreted translation, which has implications on how the rules work since many of the rules are based on the translated linguistics. Adjustment of the membership function parameters by the user through this screen can personalize the choices the system makes, and thus the system has the ability to respond with the same nuances and inflection in meaning that a population of designers can. Early evidence of this can be seen in the studies of Chapter 3. This is desirable for architects since architecture can be an art form full of meaning and nuances, and attempts should be made to provide a design environment that is free from meaning strictly imposed by others. The attempt of Figure 6.27 provides for the individuality of the designer while retaining the knowledge of the generality.

6.4 USER INTERFACE

A description of the organization of the interface and interesting interface image objects has been presented in the previous section. The interface windows attempt to organize information into physical, visual, and performance categories as well as keeping the user informed on how the design process is unfolding. It is important to understand that many of the image objects perform multiple roles apart from data capture, such as sorting, initiating functions, sending messages to other objects, etc. Not only are these objects a port to the system, but an intelligent port that can provide assistance in directing the information to an appropriate system object for response. This is the attitude taken here, thus many of the image objects are complex in themselves and form a vital contribution in

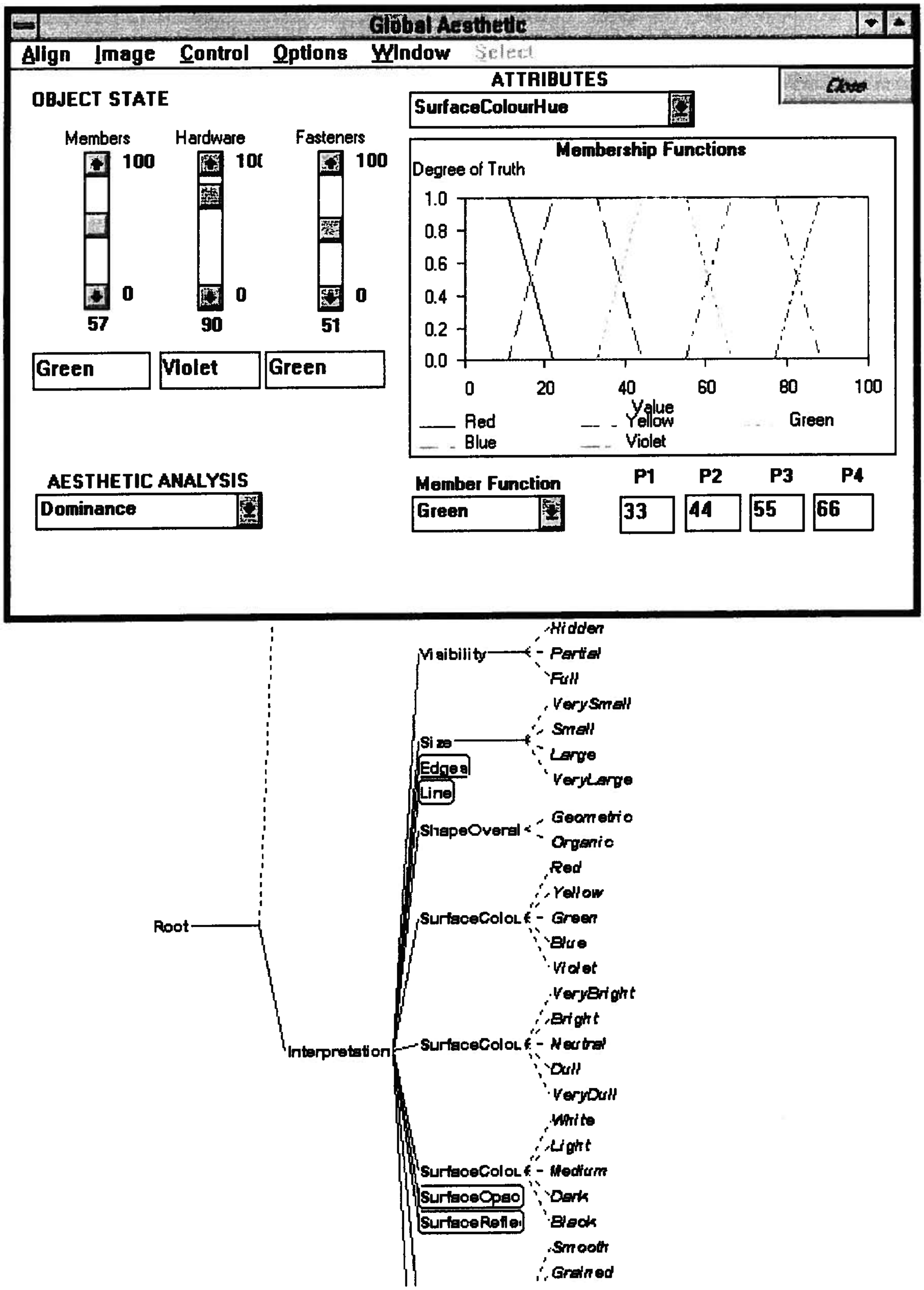


Figure 6.27 Aesthetic Controls Window

the overall system architecture. It is with this view, that the next future step should advance to direct user-engagement with on-screen design objects, just like the real world; objects which have inherent properties for structural performance, surface qualities, aesthetic parameters, and rules of usage. In this way, a more fulfilling role for the interface can be provided to the user consistent with the desirable qualities of an interface previously mentioned.

6.5 SYSTEM INTERFACE

A system interface is one that allows the easy and transparent exchange of information from one computer system to another. In the system described in Section 6.3.4, no external system interface was used for calculation as in the study of Chapter 5. The only interface provided was the indirect link between the system and the ASCII files of the Parts and Surfaces databases. In this case, the link is one-way: from the files to the system. As more data is added to the files, more data can be added to the system without altering the system's control structure. It is conceivable that, if graphics output is a desirable feature, an interface will need to be provided between the connection design application and the graphics application so that if data changes in one application, the other one is updated automatically.

6.6 SUMMARY

This section intended to demonstrate the ease of encapsulating the knowledge of connection design through the use of a real-world allegory inside a computer. This was facilitated with an object-oriented rule-based expert system developer's shell. A number of innovations in using the shell resulted including dynamic object tree creation and smart parts with a fuzzy intelligence interface. The system responds to design requirements with a variety of generated connection solutions from which the user has a choice for consideration. The implementation method is: simple in development but powerful, user friendly, and able to handle quantitative and qualitative data for analysis. The system features easy expandability by the user since the knowledge base and control are independent. The user can simply add more objects (Givers, Takers, Transferors, Distributors) as the need arises according to the database format provided. Attribute "hooks" to external CAD packages are also available to enable non-inclusive solutions to be developed in future work.

Chapter 7 VALIDATION CASES

7.1 TEST CASE ASSUMPTIONS AND LIMITATIONS

This Chapter is intended to demonstrate the principles discussed in the previous chapters by examining a few test cases. Each of the cases is presented followed by a summary of the findings in light of the approach taken in this work. The fundamental purpose of the connection design system in this thesis is to present a variety of configuration alternatives for a connection design between two timber members. The important word here is *variety*, which is something a designer, out of respect for time and the client's design budget, doesn't often have the luxury to consider. Variety allows the designer to explore more alternatives that could make the difference between a solution and a good informed solution to the problem at hand, which has obvious benefits to the designer and the client. Further, timber connections are tricky to configure because of all the background design information that needs to be assimilated, and even arriving at an initial idea can be a problem. Often and more common currently, the configuration that first comes to mind is the one that ends up being built.

Because of the emphasis on configuration, detailed positioning and capacity determination for the distributors and some hardware is not undertaken by this system. However, an extension to the system of the form presented in Chapter 5 could fill this void.

The parts database in this study was limited to connection hardware that joined two timber members together. This limitation is not a severe one, since hardware that joins multiple members can be added to the database. The hardware surface description technique is the same requiring only the surface identity of the additional Givers and Takers connected by the hardware. This can be incorporated easily in future versions of the program since the object description for the Transferors in the program is already multi-surfaced.

In order to investigate the value of the system in configuring timber connections, some connection designs for timber structures were studied, and then given to the system to make design suggestions. The selected connections were taken from Goetz *et al.* (1989); a book on the current state-of-the-art in timber construction and revered by many architects and engineers as a significant source of inspiration and creativity in timber design. An implementation design example was presented in Chapter 3 to show illustratively the logic flow and information provided by the system, so only the problem set-up and results will be presented here. Each test was run using the full parts database (no preference restriction was placed on selection) as given in Appendix A.

7.1.1 Beam Splice Connection

This splice connection can occur frequently in heavy timber frames that employ the Gerber system as is common in warehouse construction. The details of such a beam connection problem are given for the "Old Warehouse" in Figure 7.1. All of the loading on the connection is predetermined and can be found from simple load path analysis and statics. Token amounts of axial load (construction movement) and eccentricity moment are also included. The desired aesthetic expression requires hiding the hardware completely, but not hiding the fasteners to show that some thought to connectivity has been given by the designer. When the problem is given to the system, the results for the configuration of the splice appear in Table 7.1. According to the hardware and fasteners stored in the database, the system has chosen to implement a concealed steel plate with various fastener possibilities. However, the need of dealing with the results of long term shrinkage of the unseasoned beams due to the constantly dry environment will require further consideration. One way is to segment the plate across the wood grain of the beams so that the movement of the wood is not unduly restrained by the fasteners penetrating the steel plate. The general suggestion of this design is consistent with that as found in Goetz *et al.* (1989:115,145). The additional configurations offered by the reference were not suggested by the system because of the limited parts and surfaces database used. The additional hardware in the reference would have to be added to the databases.

The subject of capacity was also not dealt with by the system here, which would have refined the selection order in Table 7.1 even more. However, in the next step, this could be accomplished by sending each of the selected Options through a further procedure as detailed in Chapter 5 for bolts so that capacities and

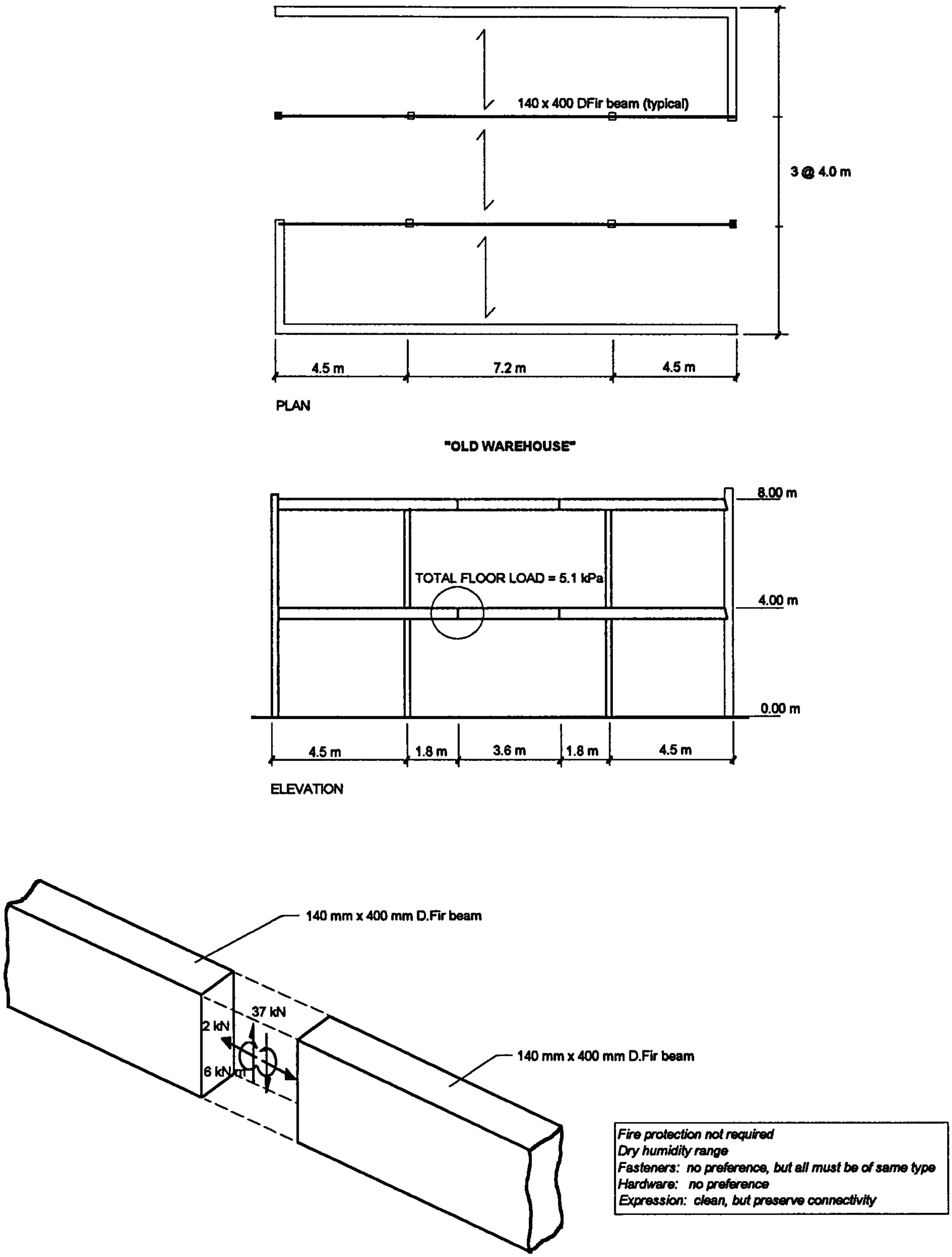
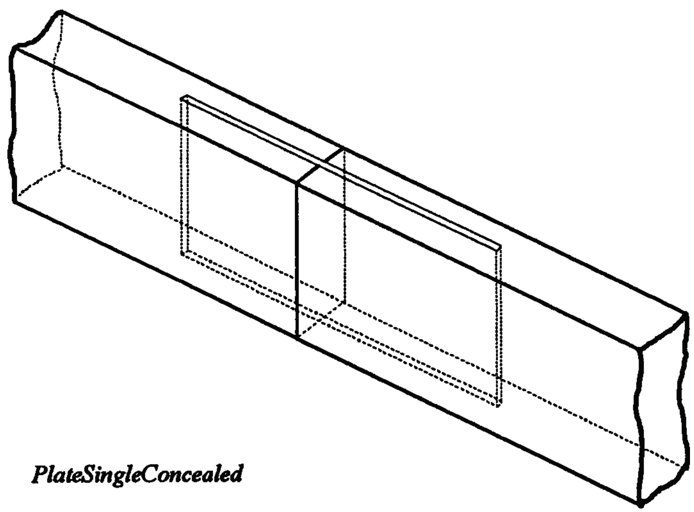


Figure 7.1 Beam Splice Connection

Table 7.1 Beam Splice Design Configuration Response from System

Option	Transferor	Distributors (Surface)	Cost	Concerns: Fire	Concerns: Shrinkage	Concerns: Movement
(1)	(2)	(3)	(4)	(5)	(6)	(7)
1	PlateSingleConcealed	LagScrews	4.17	None	Green beams may shrink	None
3	PlateSingleConcealed	Dowels	6.25	None	Green beams may shrink	None
2	PlateSingleConcealed	Bolts&Nuts	9.65	None	Green beams may shrink	None



PlateSingleConcealed

configurations of the distributors can be assessed. The demonstrative example in Chapter 5 carries the problem given here further for the bolted configuration of Option 2. It can be seen from the Chapter 5 example analysis in Figures 5.3 and 5.8 for a circular pattern using 12.5 mm bolts that, for the chosen number of fasteners (14), the splice is just over-designed. The bolt pattern with 16 bolts on a 75 mm grid shown in Figure 5.10, however, gives a capacity result that is greatly over-designed, suggesting that design modifications to reducing the number of fasteners should be considered. By altering the number and placement of fasteners, the designer is permitted to freely and quickly examine design consequences in relation to choosing fasteners of the same type within a design Option, or to choosing other design Options. This is a design freedom that currently cannot be easily offered by current manual methods.

7.1.2 Beam to Continuous Perimeter Column Connection

This connection is very typical in most types of heavy timber frames, and if the building is of a commercial use, some measure of fire protection must be offered to the connection. As is found in the literature, this typically means burying the connection hardware in the wood so that the wood's insulating properties against heat can be mobilized. The connection problem of Figure 7.2 from the "Old Warehouse" is given for the system to consider, this time with different fire and humidity requirements, and beam material (an engineered wood product). In this case, the service environment includes one of varying humidity, which is significant in altering the expansion and shrinkage of the member wood cells due to corresponding increases and decreases in relative humidity. Engineered wood products, manufactured dry through their cross-section, are known to rapidly absorb water when placed in a wet environment, dispersing quickly and uniformly through the cross section. When the wet cross-section is placed into a dry environment, the material behaves not unlike green timber. Connection hardware needs to be chosen in such a way that the timber cross-sections are not restricted in changing size, otherwise undesirable splitting and degradation of the wood in the vicinity of the connection will occur that may lead to degradation in the load carrying capacity of the connection. The system response for connection configurations is shown in Table 7.2. Six hardware selections were made: none, split rings, shear plates, a concealed plate, a welded T hanger, and a welded centre gusset hanger. Each of the hardware selections was mated with fasteners that could be countersunk and plugged. In the case of no hardware, the fasteners are

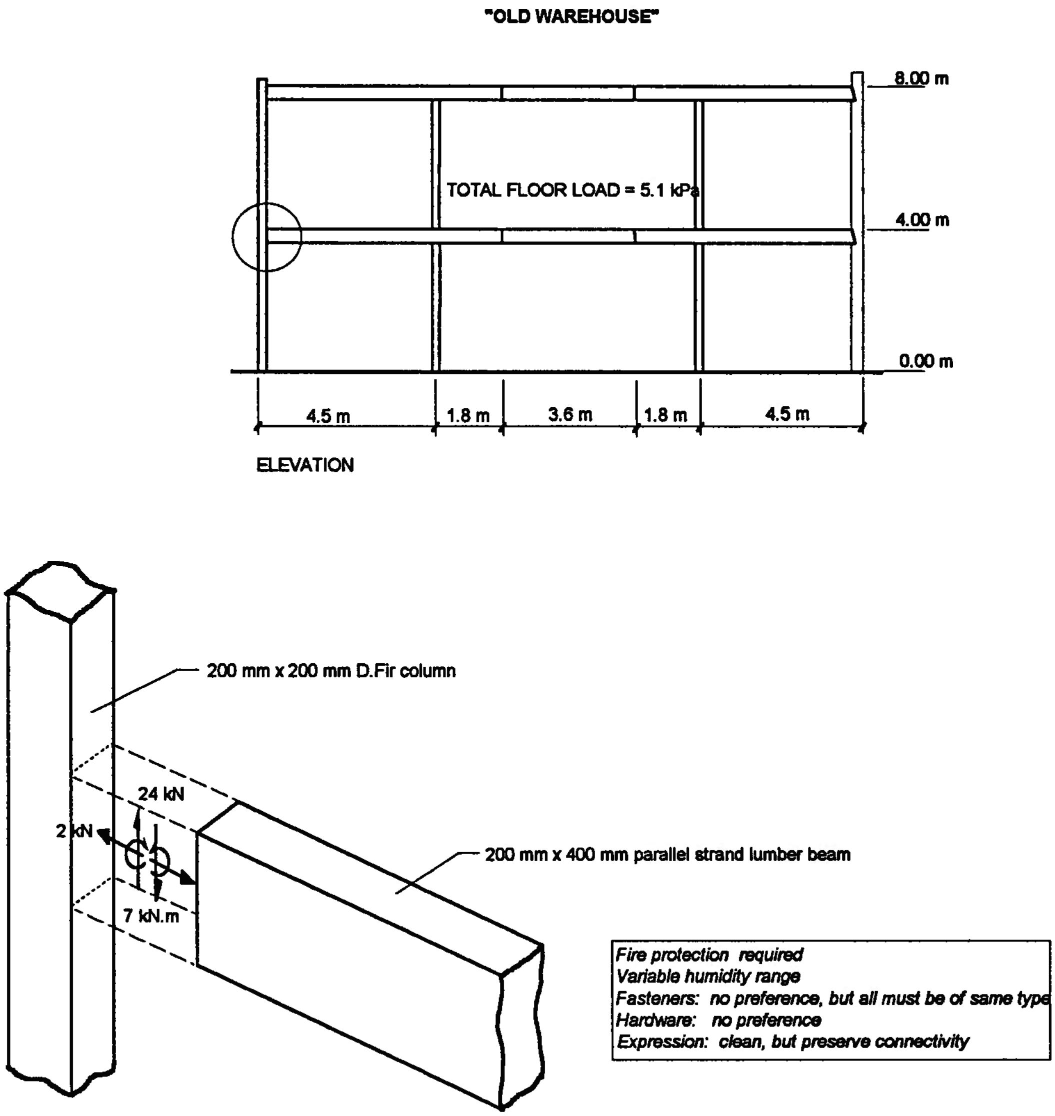


Figure 7.2 Beam to Continuous Column Connection

installed on an angle through the cross section such that their shanks cross. In one instance, the system noted a redundancy of fasteners to secure member movement was evident (a fastener selection over-subscription) and penalized the choice according to the demerit rating system as mentioned in Chapters 2 and 3.

Given the above configuration options; the best economical choice of lag screws, nails or bolts requires further consideration which is the product of number of fasteners and unit cost. For example, to determine the number and placement of fasteners, a routine similar to TimberCon (Chapter 5) can be called to detail a connection configuration as selected by the user from the tabulated configuration Options. This technique was illustrated in Section 7.1.1. New costs for each Option can be determined that better reflect the number of fasteners. The order of the economically ranked configuration Options in Table 7.2 will then shift slightly on re-sorting since more accurate costs of the respective Options are known. The designer is now in a better position to evaluate Option choices from the Table if economics plays an important role in the design requirements, and a holistic design solution can be selected with confidence. The chosen configurations of Table 7.2 are consistent with those suggested by Goetz *et al.* (1989: 85). Again, the benefit here to the designer is that a reasoned choice can be made from a number of connection possibilities that saves time and cost.

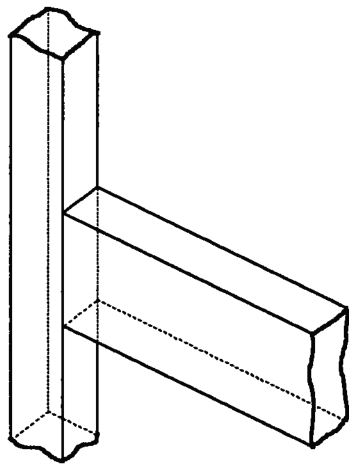
Table 7.2 Beam to Column Design Configuration Response from System

Option	Transferor	Distributors			Cost	Concerns		
		a	b	c		Fire	Shrinkage	Movement
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
16	None	Nail	Nail		0.50	Satisfied*	Minor instability **	Satisfied
17	None	LagScrew	LagScrew		0.58	Satisfied*	Minor instability **	Satisfied
1	PlateSingleConcealed	Nail	Nail		4.09	Satisfied*	Minor instability **	Satisfied
2	PlateSingleConcealed	LagScrew	LagScrew		4.17	Satisfied*	Minor instability **	Satisfied
14	SplitRing	LagScrew	LagScrew		4.19	Satisfied*	Minor instability **	Satisfied
7	HangerWeldedT	Nail	none	Nail	5.45	Satisfied*	Minor instability **	Satisfied
8	HangerWeldedT	Nail	Nail	none	5.45	Satisfied*	Minor instability **	Satisfied
15	ShearPlates	LagScrew	LagScrew		5.52	Satisfied*	Minor instability **	Satisfied
10	HangerWeldedT	LagScrew	none	LagScrew	5.53	Satisfied*	Minor instability **	Satisfied
11	HangerWeldedT	LagScrew	LagScrew	none	5.53	Satisfied*	Minor instability **	Satisfied
4	HangerWeldedCentreGusset	Nail	Nail		6.00	Satisfied*	Minor instability **	Satisfied
18	None	Bolt&Nut	Bolt&Nut		6.06	Satisfied*	Minor instability **	Satisfied
5	HangerWeldedCentreGusset	LagScrew	LagScrew		6.08	Satisfied*	Minor instability **	Satisfied
9	HangerWeldedT	Nail	Nail	Nail	5.70	Satisfied*	Minor instability **	Redundant ***
12	HangerWeldedT	LagScrew	LagScrew	LagScrew	5.82	Satisfied*	Minor instability **	Redundant ***
3	PlateSingleConcealed	Bolt&Nut	Bolt&Nut		9.65	Satisfied*	Minor instability **	Satisfied
13	HangerWeldedT	Bolt&Nut	Bolt&Nut	none	11.01	Satisfied*	Minor instability **	Satisfied
6	HangerWeldedCentreGusset	Bolt&Nut	Bolt&Nut		11.56	Satisfied*	Minor instability **	Satisfied

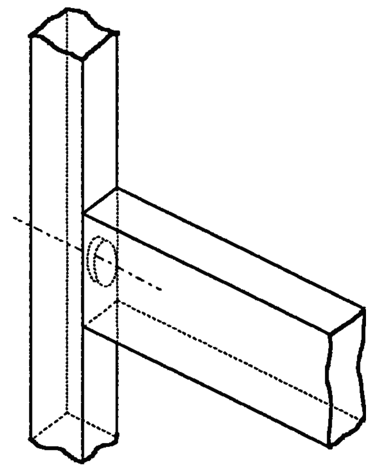
* by visual requirements.
** due to varying humidity.
*** fasteners for movement control.

continued

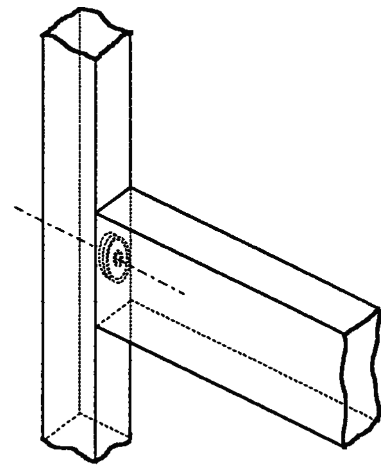
Table 7.2 *continued*



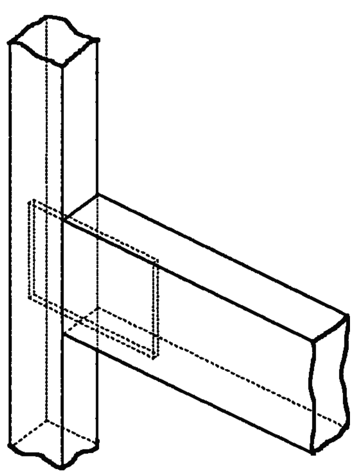
None



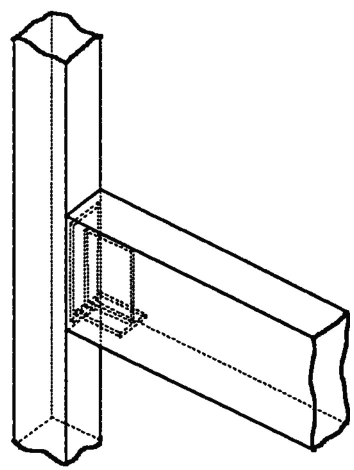
SplitRing



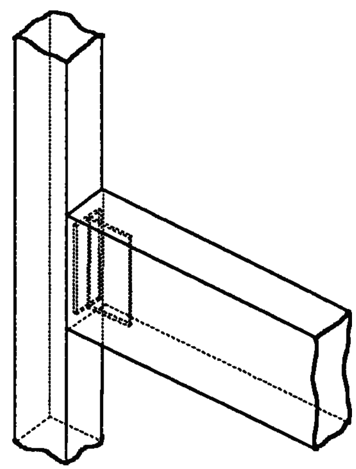
ShearPlate



PlateSingleConcealed



HangerWeldedT



HangerWeldedCentreGusset

7.1.3 Skewed Rafter into Valley Beam Side Connection

This connection is typical in timber roof framing, and can occur in buildings with complex geometric forms. The use of an automated solution to connection design is especially beneficial in the latter case since a current trend in timber architecture involves very complex forms and framing geometries, which places a lot of pressure on the innovative capacity and ingenuity of the designer to suggest reasonable connection configurations. The interior exposed roof valley beam connection problem presented in Figure 7.3 doesn't require fire protection, but full

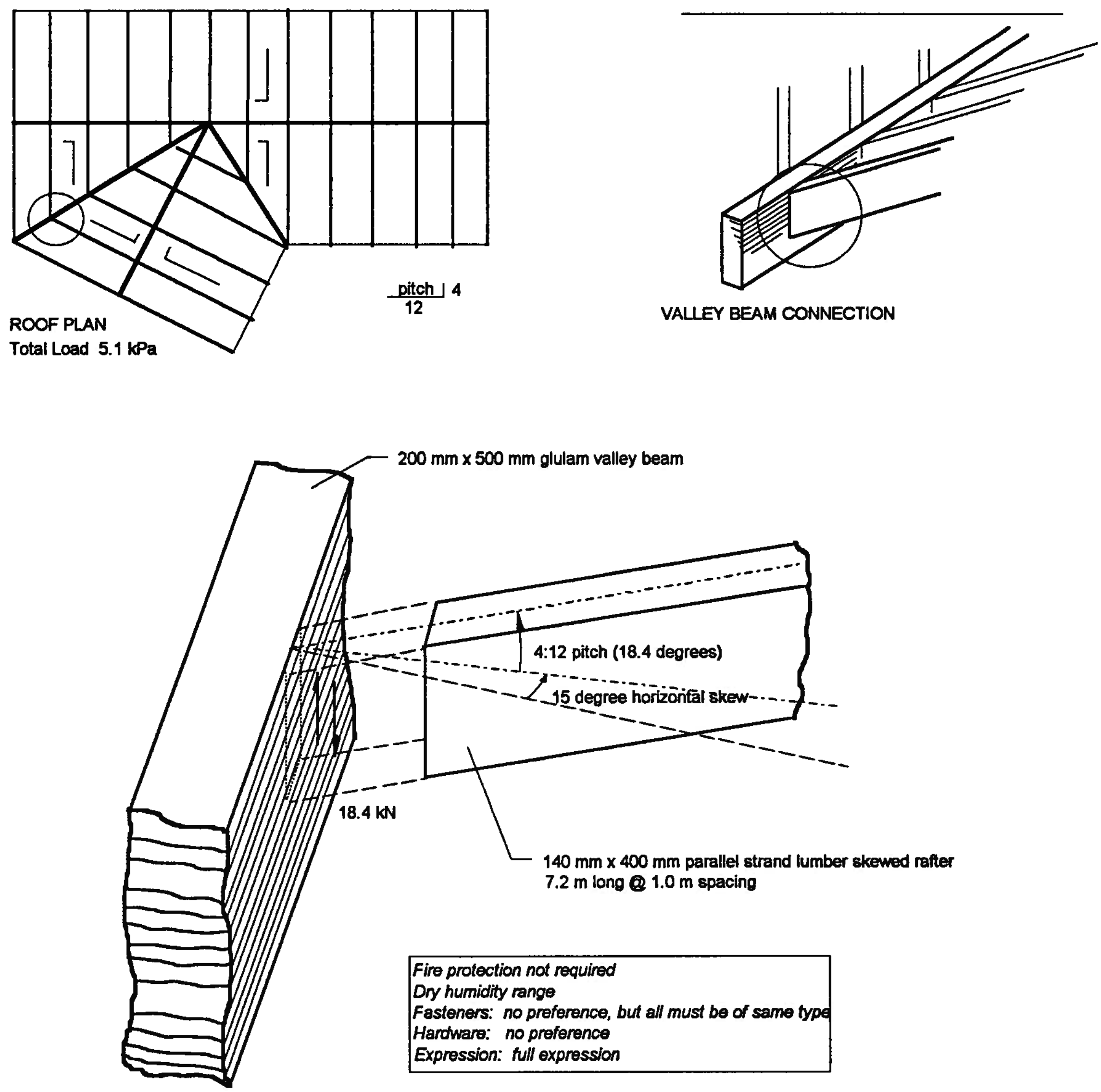


Figure 7.3 Beam to Girder Side Connection

Table 7.3 Skewed Rafter to Valley Beam Design Configuration Response from System

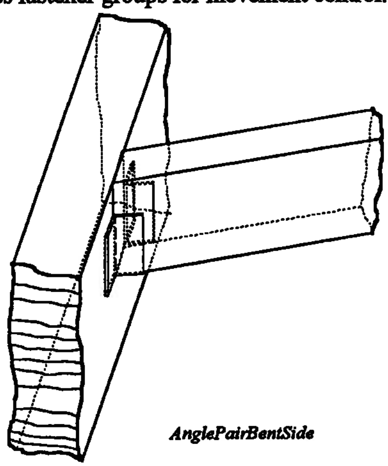
Option	Transferor	Distributors			Cost	Concerns		
		a	b	c		Fire	Shrinkage	Movement
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
16	PlatePairNarrowSide	LagScrew	LagScrew		3.08	Not an issue	Satisfied	Satisfied
17	PlatePairNarrowSide	GlulamRivets	GlulamRivets		3.56	Not an issue	Satisfied	Satisfied
1	AnglePairClipSide	LagScrew	LagScrew		4.53	Not an issue	Satisfied	Satisfied
6	AnglePairBentSide	LagScrew	LagScrew		4.75	Not an issue	Satisfied	Satisfied
2	AnglePairClipSide	GlulamRivets	GlulamRivets		5.01	Not an issue	Satisfied	Satisfied
11	AngleWeldedPairClipSide	LagScrew	LagScrew		5.12	Not an issue	Satisfied	Satisfied
7	AnglePairBentSide	GlulamRivets	GlulamRivets		5.23	Not an issue	Satisfied	Satisfied
12	AngleWeldedPairClipSide	GlulamRivets	GlulamRivets		5.60	Not an issue	Satisfied	Satisfied
21	PlatePairT	LagScrew	LagScrew		5.64	Not an issue	Satisfied	Satisfied
26	HangerWeldedLSidePlates	LagScrew	none	LagScrew	6.03	Not an issue	Satisfied	Satisfied
27	HangerWeldedLSidePlates	LagScrew	LagScrew	none	6.03	Not an issue	Satisfied	Satisfied
22	PlatePairT	GlulamRivets	GlulamRivets		6.12	Not an issue	Satisfied	Satisfied
26	HangerWeldedLSidePlates	GlulamRivets	none	GlulamRivets	6.51	Not an issue	Satisfied	Satisfied
27	HangerWeldedLSidePlates	GlulamRivets	GlulamRivets	none	6.51	Not an issue	Satisfied	Satisfied
28	HangerWeldedLSidePlates	LagScrew	LagScrew	LagScrew	6.32	Not an issue	Satisfied	Redundancy *
31	HangerWeldedLSidePlates	GlulamRivets	GlulamRivets	GlulamRivets	7.04	Not an issue	Satisfied	Redundancy *
19	PlatePairNarrowSide	Bolt&Nut	Bolt&Nut		8.56	Not an issue	Satisfied	Satisfied
4	AnglePairClipSide	Bolt&Nut	Bolt&Nut		10.01	Not an issue	Satisfied	Satisfied
9	AnglePairBentSide	Bolt&Nut	Bolt&Nut		10.23	Not an issue	Satisfied	Satisfied
14	AngleWeldedPairClipSide	Bolt&Nut	Bolt&Nut		10.60	Not an issue	Satisfied	Satisfied
24	PlatePairT	Bolt&Nut	Bolt&Nut		11.12	Not an issue	Satisfied	Satisfied
35	HangerWeldedLSidePlates	Bolt&Nut	Bolt&Nut	none	11.51	Not an issue	Satisfied	Satisfied
18	PlatePairNarrowSide	LagScrew	LagScrew		12.38	Not an issue	Satisfied	Satisfied
20	PlatePairNarrowSide	BoltedShearPlate	BoltedShearPlate		12.38	Not an issue	Satisfied	Satisfied
3	AnglePairClipSide	LaggedShearPlate	LaggedShearPlate		13.83	Not an issue	Satisfied	Satisfied
5	AnglePairClipSide	BoltedShearPlate	BoltedShearPlate		13.83	Not an issue	Satisfied	Satisfied
8	AnglePairBentSide	LaggedShearPlate	LaggedShearPlate		14.05	Not an issue	Satisfied	Satisfied
10	AnglePairBentSide	BoltedShearPlate	BoltedShearPlate		14.05	Not an issue	Satisfied	Satisfied
13	AngleWeldedPairClipSide	LaggedShearPlate	LaggedShearPlate		14.42	Not an issue	Satisfied	Satisfied
15	AngleWeldedPairClipSide	BoltedShearPlate	BoltedShearPlate		14.42	Not an issue	Satisfied	Satisfied
23	PlatePairT	LaggedShearPlate	LaggedShearPlate		14.94	Not an issue	Satisfied	Satisfied
25	PlatePairT	BoltedShearPlate	BoltedShearPlate		14.94	Not an issue	Satisfied	Satisfied
32	HangerWeldedLSidePlates	LaggedShearPlate	none	LaggedShearPlate	15.33	Not an issue	Satisfied	Satisfied
33	HangerWeldedLSidePlates	LaggedShearPlate	LaggedShearPlate	none	15.33	Not an issue	Satisfied	Satisfied
36	HangerWeldedLSidePlates	BoltedShearPlate	none	BoltedShearPlate	15.33	Not an issue	Satisfied	Satisfied

continued

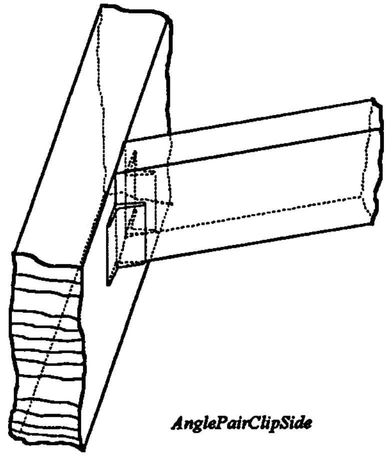
Table 7.3 *continued*

37	HangerWeldedLSidePlates	BoltedShearPlate	BoltedShearPlate	none	15.33	Not an issue	Satisfied	Satisfied
34	HangerWeldedLSidePlates	LaggedShearPlate	LaggedShearPlate	LaggedShearPlate	20.27	Not an issue	Satisfied	Redundancy *
38	HangerWeldedLSidePlates	BoltedShearPlate	BoltedShearPlate	BoltedShearPlate	20.27	Not an issue	Satisfied	Redundancy *

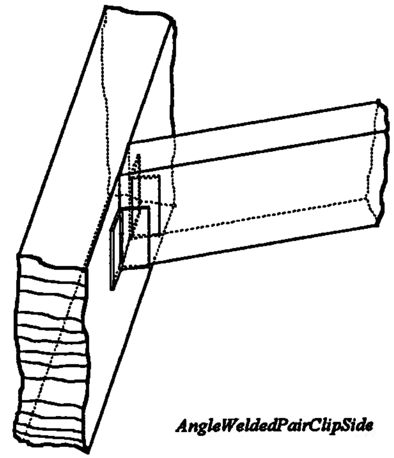
* due to excess fastener groups for movement control.



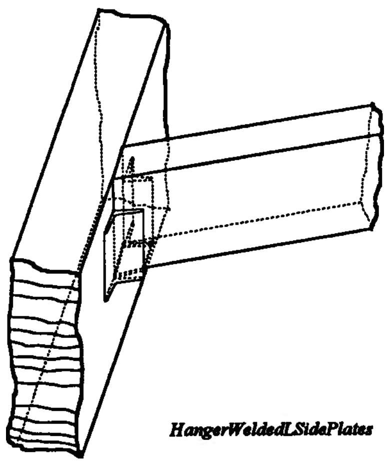
AnglePairBentSide



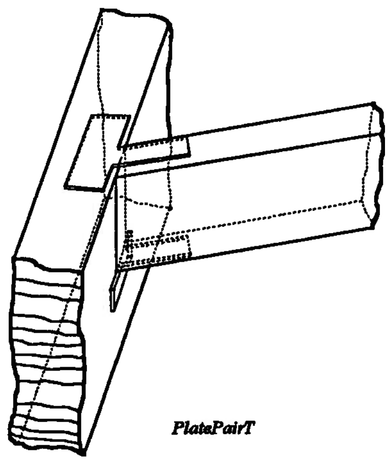
AnglePairClipSide



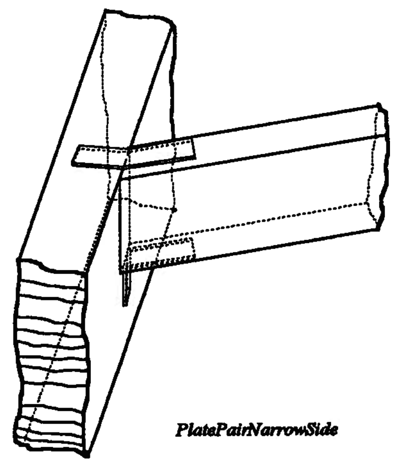
AngleWeldedPairClipSide



HangerWeldedLSidePlates



PlatePairT



PlatePairNarrowSide

connection expression, and is located in a dry service environment. Full expression implies complete visibility of all connection hardware and fastening devices. A constant dry environment is naturally beneficial for members made from engineered wood products for member cross-section stability. Similarly, a constantly wet environment is naturally beneficial to most heavy timbers of solid wood since these timbers remain in a green condition unless placed in a constantly drier environment where they can take up to a year or more depending on cross section size and humidity differential to dry out completely. The benefit is not true however, for large member cross sections made as engineered wood products such as glulam, parallel strand lumber, or laminated veneer lumber placed in wet environments since these materials are dried to very low moisture contents at manufacture, and remain so after. Moisture uptake can dramatically affect the cross section size of these products detrimentally when placed in a constantly wet service environment. In any event, the choice of hardware and fastenings must reflect the in-service environment consideration.

The connection configurations suggested by the system for the problem are given in Table 7.3. The system has selected a variety of hardware pieces that help to express the connection. Each of the hardware options also has a corresponding array of possible fastener possibilities. The fastener possibilities can be further refined through the use of a fastener capacity routine similar to that of Chapter 5 (TimberCon) to arrive at a more meaningful cost/benefit ranking. Redundancy in the placement of fasteners for a given hardware piece exacts a demerit rating price which is evident in the cost column of the Table. Small fasteners such as screws, nails, and dowels are also missing from the fastener selection since the system chose to only select fasteners of large head prominence that will help in the full expression of connectivity in perhaps a brutal style. However, the chosen configurations here are consistent with those suggested by Goetz *et al.* (1989: 129). Again, the benefit here to the designer is that a reasoned choice of an appropriate design solution can be made from a number of connection possibilities that saves time and cost.

7.1.4 Crown Hinge Connection

Again, this connection is typical in timber roof framing, and can occur in buildings with complex geometric forms, or arched structures. The rafter connection problem for the indoor swimming pool presented in Figure 7.4 doesn't require fire protection, but a very clean expression, and is located in a wet service environment. A clean expression can imply hidden connection hardware and fastening devices. A constantly wet environment can be

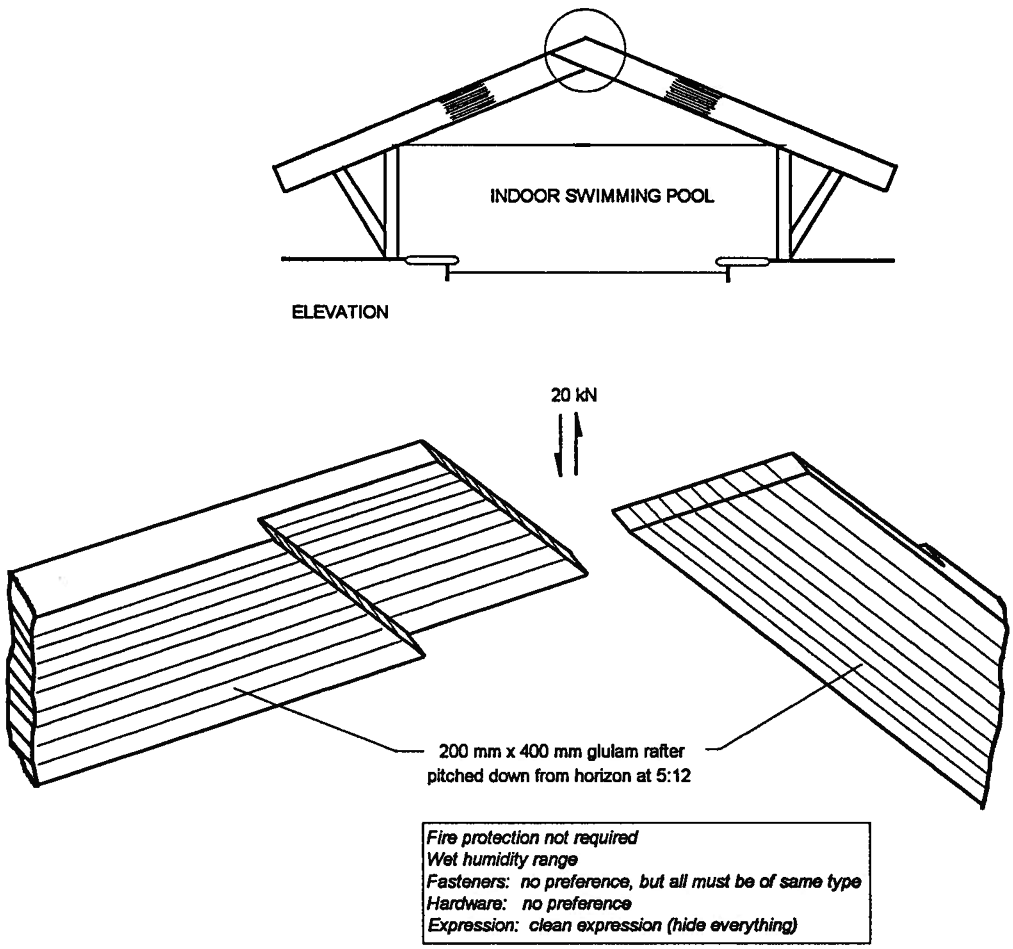


Figure 7.4 Lapped Crown Hinge Connection

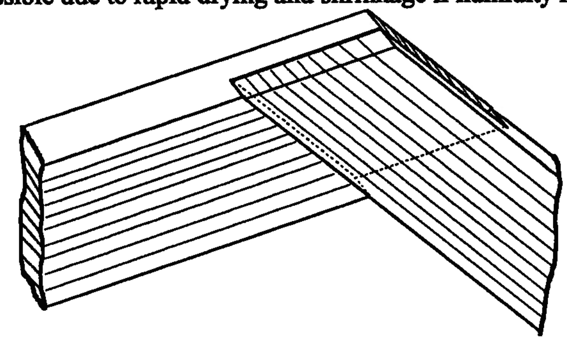
detrimental to most newly manufactured engineered wood products without prevention since the dry wood in the product very readily absorbs moisture. Once wet, engineered wood products behave much like large section green timber. Green timbers can split when placed in a constant severely dry environment that induces rapid drying and shrinking of the surface wood cells of the timber while the timber core remains in a wet green condition. The issue of drying rate has a significant effect on connections for solid wood timbers. In any event, the choice of hardware and fastenings must reflect this consideration.

The connection configurations suggested by the system for the lapped connection problem are given in Table 7.4. The system has selected a variety of hardware pieces that help to express the connection. Each of the hardware options also has a corresponding array of possible fastener possibilities. The array of possible practical fastener choices to suit the hardware can be further refined through the use of a fastener capacity routine similar to that of Chapter 5 (TimberCon) to arrive at a more meaningful cost/benefit ranking. The Birdsmouth suggestion is a timber joinery option that exists in the parts database for side-side connections, and is meant to reflect the cutting out of the wood in order to lap the members. However, the chosen configurations here are consistent with those suggested by Goetz *et al.* (1989: 135).

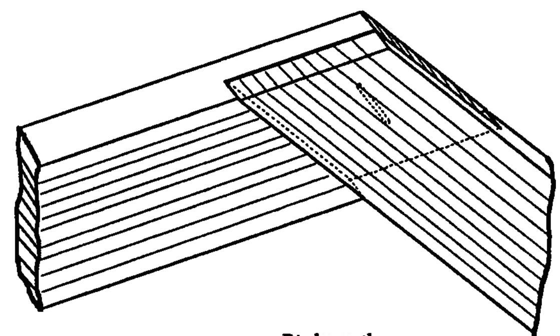
Table 7.4 Lapped Crown Hinge Design Configuration Response from System

Option	Transferor	Distributors			Cost	Concerns		
		a	b	c		Fire	Shrinkage	Movement
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
6	None	Nail	Nail		0.50	Not an issue	Shrinkage *	Satisfied
8	None	BuriedDowel	BuriedDowel		2.66	Not an issue	Shrinkage *	Satisfied
7	None	Bolt&Nut	Bolt&Nut		6.06	Not an issue	Shrinkage *	Satisfied
5	None	Adhesive	Adhesive		7.06	Not an issue	Shrinkage *	Satisfied
1	SplitRing	Bolt&Nut	Bolt&Nut		9.67	Not an issue	Shrinkage *	Satisfied
2	ShearPlates	Bolt&Nut	Bolt&Nut		11.00	Not an issue	Shrinkage *	Satisfied
4	Birdsmouth	BuriedDowel	BuriedDowel		20.58	Not an issue	Shrinkage *	Satisfied

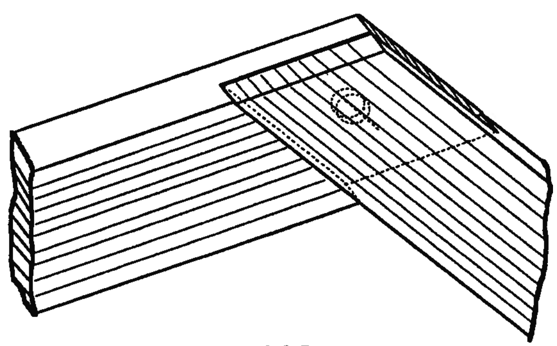
* splitting damage possible due to rapid drying and shrinkage if humidity is constantly extremely dry.



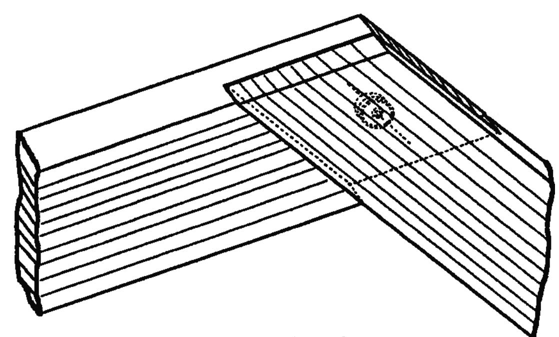
None



Birdsmouth



SplitRing



ShearPlates

If the same problem in Figure 7.4 required a butted crown hinge formed by mating the rafter ends, as in Figure 7.5, as opposed to lapping them, then the system would respond with the suggestions in Table 7.5. Again, the chosen configurations are consistent with those suggested by Goetz *et al.* (1989: 135), only with a little more variety on fasteners.

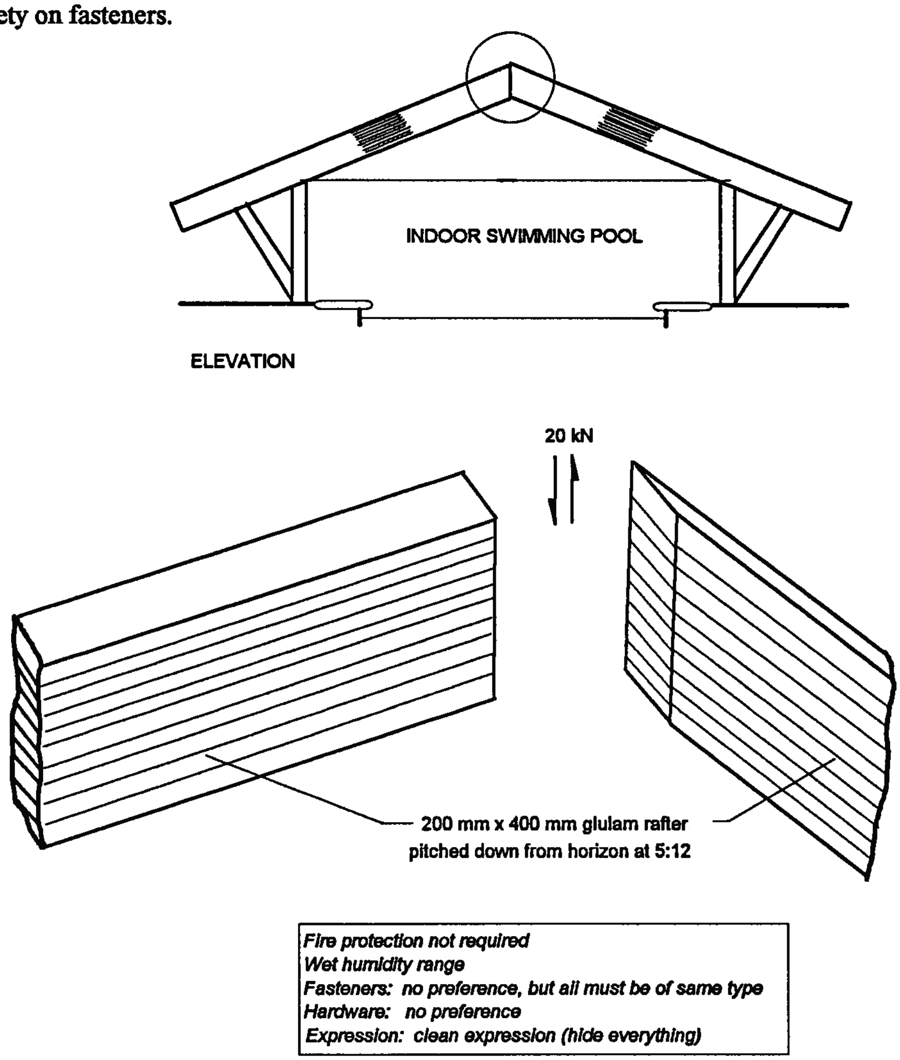
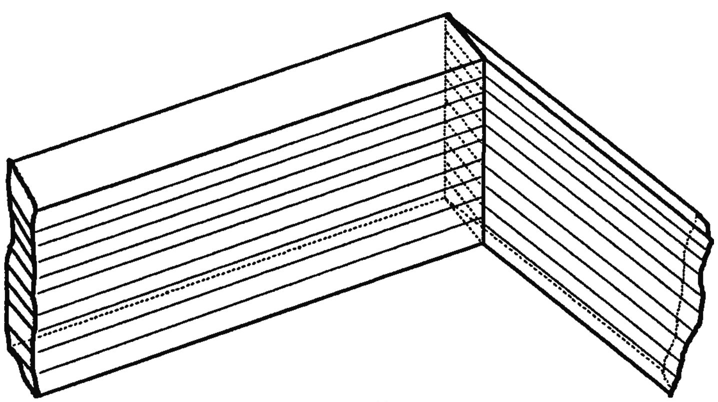


Figure 7.5 Butted Crown Hinge Connection

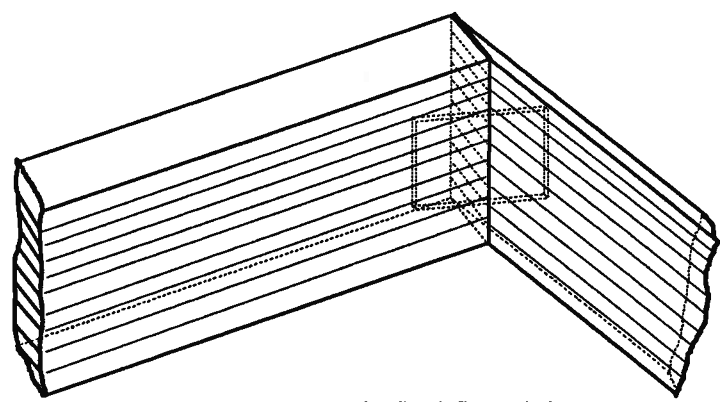
Table 7.5 Butted Crown Hinge Design Configuration Response from System

Option	Transferor	Distributors			Cost	Concerns		
		a	b	c		Fire	Shrinkage	Movement
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
2	PlateSingleConcealed	Nail	Nail		4.04	Not an issue	Shrinkage *	Satisfied
4	PlateSingleConcealed	BuriedDowel	BuriedDowel		6.25	Not an issue	Shrinkage *	Satisfied
5	None	Adhesive	Adhesive		7.06	Not an issue	Shrinkage *	Satisfied
3	PlateSingleConcealed	Bolt&Nut	Bolt&Nut		9.65	Not an issue	Shrinkage *	Satisfied
1	PlateSingleConcealed	Adhesive	Adhesive		10.65	Not an issue	Shrinkage *	Satisfied

* splitting damage possible due to rapid drying and shrinkage if humidity is constantly extremely dry.



None



PlateSingleConcealed

The suggestions in the Table are those that resolved the movement issue so that the members will not pull apart or move with respect to each other in an unstable manner. For example, making joints with buried dowels through the member end surfaces prevents movement across the member cross-section, but permits freedom of the member ends to separate as in the case of an uplift force at the hinge. This event is one that is likely to occur in strong winds on building roof peaks. All the fastener suggestions given in the table are those that can be hidden completely by countersinking and plugging the fastener heads.

7.2 SUMMARY

This Chapter intended to demonstrate the principles discussed in the previous chapters by examining a few test cases. The fundamental purpose of the connection design system in this thesis is to present a variety of configuration alternatives for a connection design between two timber members. Because of the emphasis on configuration, detailed positioning and capacity determination for the distributors and some hardware is not undertaken by this system. However, an extension to the system of the form presented in Chapter 5 (TimberCon) could fill this void. The parts database in this study was limited to connection hardware that joined two timber members together. This limitation is not a severe one, since hardware that joins multiple members can be added to the database. The hardware surface description technique is the same, requiring only the surface identity of the additional Givers and Takers connected by the hardware. This can be incorporated easily in future versions of the program since the object description for the Transferors in the program is already multi-surfaced.

In order to investigate the value of the system in configuring timber connections, some connection designs for timber structures were studied, and then given to the system to make design suggestions. The system responses for the selected connections agreed well with those taken from Goetz *et al.* (1989), with the system suggesting an even wider variety of connection possibilities in most cases.

Chapter 8 CONCLUSIONS

8.1 SUMMARY

Resolution of the problem of automating holistic connection design has resulted in a design model that is allegorical to the real world, and responds to design requirements with a variety of connection solutions. The method is simple in its development but powerful in its function in the way that it deals with the volume and diversity of information required for connection design. The method's user-friendly nature allows the handling of quantitative (numerical) and qualitative (linguistic) data for analysis. A key feature is user expandability: the knowledge base and process control are independent, allowing the user to easily add knowledge simply by adding more objects (Givers, Takers, Transferors, Distributors) with few attributes into a text file. The distinguishing characteristics of this design method over others are: ease of expansion, a holistic design approach, qualitative analysis (aesthetics), and a generative multiple solution strategy.

8.2 RELEVANCE

The relevance of such an automated design method for connections results in several beneficial gains. First, designer anxiety and design costs are reduced because the drudgery and time component are dramatically reduced. Second, an increase in design quality and efficiency should result since the designer can consider more connection design alternatives while reducing the chance of obvious or potential design errors. A new creativeness results for some designers that previously never gave aesthetics much thought. These advantages should result in an overall increase in connection design quality for the built environment.

The value of the timber connection design expert system increases the innovative design capabilities and understanding of timber connections of the architect, and the structural safety and material performance evaluation capabilities for the connection of interest to the engineer. The intuition of the architect in the geometric arrangement of connections (for aesthetics and form) is merged with the analytic assessment of structural performance which is the domain of the engineer. In a sense, the system entices movement of the two professions toward each other in a plausible contextual way; not an overlap, but a meeting. In this manner, perhaps the roles of the architect and the engineer can be better understood.

In brief, the new contribution offered by this research is the presentation of a KBES methodology for the development of a commercially viable KBES for the holistic design of structural connections in general, and timber connections in particular, with the following features:

- a very flexible and intuitive timber connection design tool for the architect that can be used to expertly check the practicality of innovative timber connection designs,
- a state-of-the-art analysis tool for the structural engineer that expertly evaluates the structural safety and adequacy, fire resistance, and economics of a timber connection design,
- a modular software architecture which can easily be updated, expanded, or used to supplement other currently existing software modules for beam, column, and structural form design,
- a formalization of knowledge in structural and architectural analysis, and design of connections, leading to a better understanding of the design process,
- a savings in cost and time realized by a new ease in connection design,
- a tool that not only enables better communication and understanding between architect and engineer, but serves as an educator for students of structural design and architecture helping them to gain expertise in a complex and challenging field.

It is anticipated that the fruits and industrial significance of the above work will shed some light on holistic design and stimulate innovative timber connection design by architects and engineers that will open up new opportunities for the use of timber in buildings or other structures.

8.3 CONTRIBUTIONS TO KNOWLEDGE

This thesis is a synthesis of a number of studies underlying a holistic approach to timber connection design. As a contribution to understanding; a descriptive background of the design process and associated problems as seen by the professions of architecture and engineering, supplemented by an extensive classified bibliography, is presented in an effort to synthesize and formalize this information. A number of new contributions to knowledge have developed from the applied research undertaken within the scope of this work which can be categorized as follows:

8.3.1 Artificial Intelligence

- *A qualitative-quantitative information translation procedure adopted from fuzzy logic membership functions for reasoning in a linguistic space was developed.* Little work has apparently been done to date in this area, but this work is particularly useful in the linkage of numerically-described information with linguistically-described information, particularly if the linguistic descriptors are considered to be vague in nature. Computer design models are often stored numerically, while in reality, design model attributes and meanings for the beholder are often described linguistically. This contribution, supported by a study (developed for computer) on proximity and another on colour, holds the promise of bridging the numerical knowledge of the engineer and the linguistically descriptive knowledge of the architect inside a computer.
- *Object-oriented technology serves as the meeting place of the two professions inside a computer in that it can well represent real-world behaviour of structural and building components.* The developed timber connection configuration system makes use of a hierarchical model that is based on real-world objects. A description of this approach and how the various objects are represented is offered. This contribution presents a practical and reasonable approach to modelling connection objects as close to the more familiar real-world objects as possible so that engineering and architectural issues can be addressed in design.

8.3.2 Architecture

- *A method that can generate design configuration alternatives for consideration based on a set of aesthetic (qualitative) and performance (quantitative) design requirements was advanced.* Architects in practice, given the luxury of time, like to explore design alternatives, while having some confidence that their innovations are practical and safe before design production work and detailing begins. This contribution points to an automated design approach that is *generative* in strategy in order to permit exploration of a number of valid design possibilities. Thus, a designer is not constrained by a prescribed design process, but freed by a number of generated design possibilities that can open up a designer's innovative capacities even further.

- *Quantitative-qualitative translation facilitates reasoning about architectural work that can be described linguistically.* A simple approach was offered for performing an aesthetic assessment of designed artifacts based on linguistic descriptors for object attributes. This contribution suggests that automated reasoning procedures can be explored that generate or analyze for a desired aesthetic and semantic for architectural designs, information that is valuable to engineers or architects at any level.
- *An approach to the aesthetic assessment of designed timber connections was advanced.* The approach was based on simple aesthetic concepts for timber connections as found in the literature.

8.3.3 Engineering

- *A method of modelling the design and behaviour of structural joints based on the load path through connection objects was developed, along with a novel approach for the assessment of relative movement of connected members.* The configuration model can be applied to any structural joint, but is developed here for timber connections. This contribution illustrates a method to configure connection objects such that load path and serviceability concerns such as movement addressed. In the case of movement, a novel approach is advanced that employs a special surface with a coding scheme to check relative movement of the connected member objects against safety and design requirements. The special surface and movement code are attributes of the hardware objects used in the connection, and can easily be developed for any piece of connection hardware. Both the load path model and the movement assessment are important developments for selection of object configuration possibilities for a connection design.
- *Modelled objects should correspond to the real-world counterparts in their behaviour and ability to visually represent themselves on a computer monitor.* Very limited work was performed on the graphical representation of computer objects for connections as development of such a production system is already commerically viable and was considered outside of the scope of this thesis. However, it was recommended that future systems for automated connection design contain facilities for direct on-screen engagement and

manipulation of the objects within the designed artifact with instant feedback from the system on the value of a design move aesthetically and practically.

- *Spreadsheet analysis engines tied to an object-oriented development was investigated.* A pilot study was undertaken in the form of developed computer application that determines the correct placement and strength capacity of a timber connection using bolts. The development linked an object-oriented connection representation model with a spreadsheet analysis engine. For the application in this research, the convenience of programming and establishing a link to an object-oriented environment was found to be offset by high software overhead and spreadsheet iterative calculation problems using Dynamic Data Exchange. The use of simple high-level language program executables to serve as analysis engines, and data passage between executables and the object-oriented environment through temporary data files, is currently recommended particularly if recursive or iterative calculations are involved.
- *A brief synthesis of linguistically-based rules of thumb for various practical concerns in the design of timber connections was presented.* These few rules, useful to timber connection designers, can serve as the beginning of the creation of a larger set of heuristics for timber connection design and are presented in flowchart form.

8.4 FUTURE DEVELOPMENTS

The scope and intention of this research is the initial stepping stone to laying an alternative foundation for more holistic connection design tools. While this study concentrated on timber connections, it is conceivable to extend this method to other modular connection systems such as steel, or precast concrete; and to other scales from children's building toys, to furniture, to buildings. The immediate work requires extension to interface with CAD programs so that enhanced graphical I/O is achieved, as design is predominantly visual. The importance of the present work was to lay out the means of automated holistic connection design in order for this to happen.

Another interesting area for exploration is in the creation of perception membership functions for designed object attributes and expressions. The present work formed the foundation for this idea backed by two studies. More work is needed here, especially in capturing the "feelings" of people in response to changes in various artifact attributes. This research may lead to a better understanding of how people relate to the built environment which becomes an important issue in architecture and sociology.

A final area for immediate work is the creation of simple high-level language capacity performance models for connection hardware classes. Again, the "hooks" for the performance models and an approach to implementation are already in place in the current approach, and the extended work involved would essentially be programming in nature.

References

Abella, A.; Kender, J.R. (1993): *Qualitatively Describing Objects Using Spatial Prepositions*, Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI '93), American Association for Artifical Intelligence, July 11-15, 1993, Washington, D.C., AAAI Press / The MIT Press, Cambridge, Mass: 536-540.

Adeli, H. (1988): *Interactive Microcomputer-Aided Structural Steel Design*, Prentice-Hall, Englewood Cliffs, NJ.

Akin, O. (1986): *Psychology of Architectural Design*, Pion Limited, London, UK.

Archea, J. (1987): *Puzzlemaking: What Architects Do When No One is Looking*, in Kalay (1987a) :37-52.

Aristotle (1957): *De Anima*, Loeb Classical Library, Harvard University Press, Cambridge, Mass.

Autodesk (1994): *AutoCAD Designer*, Autodesk Inc..

Bazjanac, V. (1974): *Basic Questions of Design Theory*, Chapter: Architectural Design Theory: Models of the Design Process, Spillers, W.R., North-Holland Publishing Co., Amsterdam, The Netherlands.

Benjamin, B.S. (1984): *Structures for Architects*, Van Nostrand Reinhold, New York.

Born, R.P. (1987): *Artifical Intelligence: The Case Against*, Croom Helm Ltd., Beckenham, Kent, UK.

Building Arts Forum (1991): *Bridging the Gap: Rethinking the Relationship of Architect and Engineer*, Proceedings of the Building Arts Forum / New York Symposium - April 1989, Van Nostrand Reinhold, New York.

Canadian Institute of Steel Construction (1992): *Handbook of Steel Construction*, 5th Ed., CISC, Willowdale, Ontario.

Canadian Wood Council (1990): *Wood Design Manual*, Canadian Wood Council, Ottawa.

Canadian Wood Council (1991): *Wood Reference Handbook*, Canadian Wood Council, Ottawa, Canada.

Carmichael, E.N. (1984): *Timber Engineering*, E. & F.N. Spon Ltd., London.

Coyne, R.D.; Rosenman, M.A.; Radford, A.D.; Balachandran, M.; Gero, J.S. (1990): *Knowledge-Based Design Systems*, Addison-Wesley Publishing Company, Sydney, Australia.

DiNenno, P.J. (1988): *SFPE Handbook of Fire Protection Engineering*, 1st Ed., Society of Fire Protection Engineers, National Fire Protection Association, Boston, MA.

Dym, C.L.; Henchy, R.P.; Delis, E.A.; Gonick, S. (1988): **A Knowledge-based System for Automated Architectural Code Checking**, *Computer-Aided Design*, 20(3): 137-145.

Edel, D.H. Jr. (1967): *Introduction to Creative Design*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

Forde, B.W.R. (1989): *An Application of Selected Artifical Intelligence Techniques to Engineering Analysis*, Ph.D. Thesis, Dept. of Civil Engineering, University of British Columbia, Vancouver.

Forest Products Laboratory (1987): *Wood Handbook: Wood as an Engineering Material*, US Department of Agriculture, Washington, DC.

Ghosh, D.K.; Kalyanaraman (1993): **KBES for Design of Steel Structural Elements,** *ASCE Journal of Computing in Civil Engineering*, 7(1):23-35.

Glassner, A .S. (1990): *Graphics Gems*, Academic Press Inc., SanDeigo, California.

Goetz, K-H.; Hoor, D.; Moehler, K.; Natterer, J. (1989): *Timber Design and Construction Sourcebook*, McGraw-Hill Publishing Company, New York.

Hall, R. (1989): *Illumination and Color in Computer Generated Imagery (Monographs in Visual Computing)*, Springer-Verlag New York Inc., New York.

Hearn, D.; Baker, M.P. (1986): *Computer Graphics*, Prentice-Hall, Englewood Cliffs, New Jersey.

Holgate, A. (1992): *Aesthetics of Built Form*, Oxford University Press, New York.

Holgate, A. (1986): *The Art in Structural Design: An Introduction and Sourcebook*, Oxford University Press, New York.

IntelliCorp Inc. (1992): *KAPPA-PC*, Version 2.1 for MS-Windows 3.0, 3.1, Expert System Application Development Environment, Mountain View, California.

Isaac, A.R.G. (1971): *Approach to Architectural Design*, Butterworth & Co., Toronto, Ontario.

Kalay, Y.E. (1987a): *Computability of Design*, John Wiley & Sons, New York.

Key Papers:
- Woodbury, R.F.: **Strategies for Interactive Design Systems, 11-35.**
- Archea, J.: **Puzzle-making: What Architects Do When No One is Looking, 37-52.**
- Shapiro, S.C.; Geller, J.: **Artificial Intelligence and Automated Design, 173-187.**
- Schmitt, G.: **Expert Systems in Design Abstraction and Evaluation, 213-244.**
- Norman, R.B.: **Intuitive Design and Computation, 295-301.**
- Swerdloff, L.M.; Kalay, Y.E.: **A Partnership Approach to Computer-aided Design, 315-336.**
- Majkowshi, B.R.; Kalay, Y.E.: **Computability of Design, 349-357.**

Kalay, Y.E. (1989): *Modeling Objects and Environments*, John Wiley & Sons, New York.

Kosko, B. (1992): *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, Englewood Cliffs, NJ.

Krasojevic, M.; Haller, P. (1994): *Knowledge Based System Supporting the Design of Timber Joints,* presented at European Group for Structural Engineering Applications of Artificial Intelligence (EG-SEA-AI) First Workshop, March 21-22 1994, Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland.

Lin, T.Y.; Stotesbury, S. D. (1988): *Structural Concepts and Systems for Architects and Engineers*, Second Edition, Van Nostrand Reinhold, New York.

Maher, M.L. (1984): *HI-RISE: A Knowledge-based Expert System for the Preliminary Structural Design of High Rise Buildings*, Ph.D. Thesis, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA.

Means, R.S. (1992): *Building Construction Cost Data 1989,* 47th Annual Ed., R.S. Means Co. Ltd., Kingston, MA.

Mitchell, J.R.; Radford, A.D. (1987): **EAVE, A Generative Expert System for Detailing**, *Environment and Planning B: Planning and Design*, 14(3): 281-292.

Mitchell, W.J. (1990): *The Logic of Architecture*, The M.I.T Press, Cambridge, Mass.

Mitchell, W.J. and McCullough, M. (1991): *Digital Design Media: A Handbook for Architects and Design Professionals*, Van Nostrand Reinhold, New York.

Motorola Fuzzy Logic (1993): *Motorola Fuzzy Logic Seminar*, Notes, Austin, Texas: 1-45.

Neis, M.E.; Neis, V.V. (1993): **An Interactive Program for the Limit States Design of Wood**, *Canadian Journal of Civil Engineering*, 20(2): 299-304.

Nieminen, J.; Kanerva, J.; Mantyla, M. (1990): **Feature-based Design of Joints**, *Advanced Geometric Modelling for Engineering Applications*, Krause, F.L., Jansen,K.,Eds., Elsevier Science Publishers BV (North-Holland), Amsterdam, The Netherlands.

Oxman, R.; Gero, J.S. (1987): **Using an Expert System for Design Diagnosis and Design Synthesis**, *Expert Systems*, 4(1): 4-15.

Prak, N.L. (1968): *The Language of Architecture*, Mouton & Co., The Hague, Paris.

Radford, A.D.; Gero, J.S. (1988): *Design by Optimization in Architecture*, Building, and Construction, Van Nostrand Reinhold Company, New York.

Rychener, M.D. (1985): **Expert Systems for Engineering Design**, *Expert Systems*, 2(1):30-44.

Sibigtroth, J.M. (1992): *AI EXPERT*, Chapter: Implementing Fuzzy Logic Expert Rules in Hardware, Miller Freeman Inc., San Francisco, Calif., April.

Shack, J. (1989): *Tectonic Ways of Seeing, Thinking, Making*, Paper presented to the Association of Collegiate Schools of Architecture National Meeting, Chicago.

Sham, S.H.R. (1993): **Nonmonotonic Reasoning in Design**, *ASCE Journal of Computing in Civil Engineering*, 7(1): 36-53.

Simpson Strong-Tie Company (1993): *Electronic Catalog Software*, Simpson Strong-Tie Company Inc., Pleasanton, California.

Smith, M.H.; Takagi, H. (1993): *Optimization of Fuzzy Systems by Switching Reasoning Methods Dynamically*, Proceedings of the Fifth International Fuzzy Systems Association World Congress (IFSA '93): 1354-1357.

Stalnaker, J.; Harris, E.C. (1989): *Structural Design in Wood*, Van Nostrand Reinhold, New York.

Stefik, M. (1993): *Introduction to Knowledge Systems*, (fall 1993 draft), Morgan Kaufmann, Inc.

Stiemer, S.F.; Lo, D. (1988): **Formatted Spreadsheets for Engineers**, *Microcomputers in Civil Engineering*, 3 :145-156.

Tang, S.Y.J. (1982): **Tanguage and Architecture**, *Architecural Education and the University: Proceedings of the 70th Annual Meeting of the Association of Collegiate Schools of Architecure*, ACSA, Washington, DC: 197-225.

Taylor, R.J. (1991): *Ultimate Load Analysis of Mechanically Fastened Connections in Wood Under Static or Cyclic Loads - A Research Proposal*, (unpublished), Dept. of Civil Engineering, University of British Columbia, Vancouver.

Taylor, R.J. (1992): *TimberCon Pilot Project Demonstration of a Timber Connection Design Expert System*, (runtime computer software), Dept. of Civil Engineering, University of British Columbia, Vancouver.

Taylor, S.I. (1982): **Structural Systems Comparison without Preliminary Design,** *Architecural Education and the University: Proceedings of the 70th Annual Meeting of the Association of Collegiate Schools of Architecure*, ACSA, Washington, DC: 186-192.

Taylor, W. (1988): *What Every Engineer Should Know About Artificial Intelligence*, The M.I.T. Press, Cambridge, Mass.

Thomson, J.V.; Marksjo, B.S.; Sharpe, R. (1987): **A Knowledge-representation Language for Engineering Design Codes (TIMBER EXPERT),** *Environment and Planning B: Planning and Design*, 14(3): 293-304.

Thornton, C.H.; Tomasetti, R.; Tuchman, J.; Joseph, L. (1993): *Exposed Structure in Building Design*, McGraw-Hill, NewYork, NY.

Vallhonrat, C. (1988): **Tectonics Considered: Between the Presence and Absence of Artifice,** *Perspecta: The Yale Architectural Journal*, Rizzoli International Publications Inc., New York, 24: 123-135.

Vermeulen, A. (1993): **Object-oriented Finite-element Software,** *Dr. Dobb's Journal*, Miller Freeman Inc., San Mateo, CA, 18(6): 86-91.

Vitruvius, M.P. ( ): **De architectura,** in Morgan, M.H. (1914): *Ten Books on Architecture*, (English translation), Harvard University Press.

White, R.H. (1988): **Analytical Methods for Determining Fire Resistance of Timber Members,** Section 3, Chapter 8, in DiNenno (1988): *SFPE Handbook of Fire Protection Engineering*, SFPE, Boston, MA.

Wigan, M.R. (1987): **Legal and Ethical Issues in Expert Systems Used in Planning,** *Environment and Planning B: Planning and Design*, 14(3): 305-321.

Woodbury, R.F. (1987): **Strategies for Interactive Design Systems,** in Kalay (1987a) :11-35.

Zimmerman, H.J. (1991): *Fuzzy Set Theory - and its Applications*, 2nd Edition, Kluwer Academic Publishers, Boston, Mass.

Classified Bibliography

The following material was useful in supplying background, historical, and sometimes conflicting information to the development and preparation of this thesis.

Design Theory

Akin, O. (1978): **How Do Architects Design?**, *Artifical Intelligence and Pattern Recognition in Computer Aided Design*, Latombe, Ed., North-Holland Publishing Company, Amsterdam, The Netherlands.

Akin, O. (1986): *Psychology of Architectural Design*, Pion Limited, London, UK.

Antoniades, A.C. (1990): *Poetics of Architecture: Theory of Design*, Van Nostrand Reinhold, New York.

Alexander, C. (1964): *Notes on the Synthesis of Form*, Harvard University Press, Cambridge, Mass.

Banares-Alcantara, R. (1991): **Representing the Engineering Design Process: Two Hypotheses**, *Computer-Aided Design*, 23(9): 595-603.

Building Arts Forum / NewYork (1991): *Bridging the Gap: Rethinking the Relationship of Architect and Engineer*, Proceedings of the Building Arts Forum / New York Symposium - April 1989, Van Nostrand Reinhold, New York.

Ching, F.D.K. (1979): *Architecture: Form, Space and Order*, Van Nostrand Reinhold Company, New York.

Cross, N. (1977): *The Automated Architect*, Pion Limited, London, UK.

Gero, J.S.; Coyne, R.D. (1985): **Logic Programming as a Means of Representing Semantics in Design Languages**, *Environment and Planning B: Planning and Design*, 12: 351-369.

Isaac, A.R.G. (1971): *Approach to Architectural Design*, Butterworth & Co., Toronto, Ontario.

Kappraff, J. (1991): *Connections - The Trigonometric Bridge Between Art and Science*, McGraw-Hill Inc., New York.

Lawson, B. (1990): *How Designers Think*, 2nd Ed., Butterworth Architecture, London, UK.

Maidstone, R.J. (1975): *Developments in Structural Form*, Penguin Books, Harmondsworth, Middlesex, England.

Newsome, S.L.; Spillers, W.R.; Finger, S. (1988): *Design Theory '88*, Proceedings of the 1988 NSF Grantee Workshop on Design Theory and Methodology (2nd:1988:Troy, NY), Springer-Verlag, New York.

Norberg-Schulz, C. (1965): *Intentions in Architecture*, The M.I.T. Press, Cambridge, Mass.

Pagani, F. (1980): *Designing for Design Education*, M.E.Sc. Paper, Faculty of Environmental Studies, York University, Downsview, Ontario.

Simon, H.A. (1981): *The Sciences of the Artificial*, 2nd Ed., The M.I.T. Press, Cambridge, Mass.

Spillers, W.R. (1986): **Design Theory versus Architectural Design**, *Environment and Planning B: Planning and Design*, 13(2): 243-248.

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

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

Stiny, G.; Gips, J. (1974): **Formalization of Analysis and Design in the Arts,** *Basic Questions of Design Theory*, Spillers, W.R., Ed., North-Holland Publishing Comapny, Amsterdam, The Netherlands.

Thornton, C.; Tomasetti, R.; Tuchman, J.; Joseph, L. (1993): *Exposed Structure in Building Design*, McGraw-Hill Inc., New York.

Computer Aided Design

Adeli, H.; Fiedorek, J. (1986a): **Microcomputer-Aided Design and Drafting of Moment Resisting Connections in Steel Buildings,** *Microcomputers in Civil Engineering*, 1(1): 32-44.

Adeli, H.; Fiedorek, J. (1986b): **A MICROCAD System for the Design of Steel Connections - Program Structure and Graphic Algorithms,** *Journal of Computers and Structures*, 24(2): 281-294.

Adeli, H.; Fiedorek, J. (1986c): **A MICROCAD System for the Design of Steel Connections - Applications,** *Journal of Computers and Structures*, 24(3): 361-374.

Adeli, H; Fiedoruk, J. (1986d): **Computer-Aided Design of Beam-Column Seated Angle Connections,** *International Journal of Civil Engineering for Practicing and Design Engineers*, 5(8): 719-756.

Adeli, H; Fiedoruk, J. (1987): **Interactive Microcomputer-Aided Design of Shop-Welded and Field Bolted Beam-Column Connections,** *Computer-Aided Design*, 19(3): 115-121.

Adeli, H.; Hawkins, D. (1986): **A Graphics Preprocessor for Computer-Aided Design and Drafting of Frame Structures,** *Microcomputers in Civil Engineering*, 1(2): 107-120.

Bjork, B-C. (1989): **Basic Structure of a Proposed Building Product Model,** *Computer-Aided Design*, 21(2): 71-78.

Carrasco, R. (1988): **Three Dimensional Modelling as Part of Architectural Design,** *Computer-Aided Design*, 20(4): 217-220.

Eastman, C.M. (1989): **Architectural CAD: A Ten Year Assessment of the State of the Art,** *Computer-Aided Design*, 21(5): 289-292.

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

Ferrante, A.J.; Moriera, L.F.R.; Boggio Videla, J.M.; Montagu, A. (1991): *Computer Graphics for Engineers and Architects*, Elsevier Science Publishers BV and Computational Mechanics Publications, The Netherlands and New York.

Kalay, Y.E. (1985): **Redefining the Role of Computers in Architecture: From Drafting/Modelling Tools to Knowledge-based Design Assistants,** *Computer-Aided Design*, 17(7): 319-327.

Kalay, Y.E. (1987b): *Graphic Introduction to Programming*, John Wiley & Sons, New York.

Kalay, Y.E. (1989): *Modeling Objects and Environments*, John Wiley & Sons, New York.

Kalay, Y.E. (1992): *Evaluating and Predicting Design Performance*, John Wiley & Sons, New York.

Kharrufa, S.; Saffo, A.; Aldabbagh, H.; Mahmood, W. (1988): **Developing CAD Techniques for Preliminary Architectural Design**, *Computer-Aided Design*, 20(10): 581-588.

Maver, T.; Wagter, H. (1988): *CAAD Futures '87*, Proceedings of the Second International Conference on Computer Aided Architectural Design Futures, Eindhoven, The Netherlands, 20-22 May 1987, Elsevier Science Publishing Company Inc., Amsterdam, The Netherlands.

Meier, A. (1986): **Applying Relational Database Techniques to Solid Modelling**, *Computer-Aided Design*, 18(6): 319-326.

Ng, W.Y. (1991): **Generalized Computer-Aided Design System: A Multiobjective Approach**, *Computer-Aided Design*, 23(8): 548-553.

Roller, D. (1991): **An Approach to Computer-Aided Parametric Design**, *Computer-Aided Design*, 23(5): 385-391.

Roy, U.; Liu, C.R. (1988): **Establishment of Funcional Relationships Between Product Components in Assembly Data Base**, *Computer-Aided Design*, 20(10): 570-580.

van Wijk, J.J. (1986): **SML: A Solid Modelling Language**, *Computer-Aided Design*, 18(8): 443-449.

Rychener, M.D. (1988): *Expert Systems for Engineering Design*, Academic Press Inc., San Diego, California.
Key Papers: Rychener, M.D.: **Research in Expert Systems for Design, 1-33.**
Maher, M.L. : **HI-RISE: An Expert System for Preliminary Structural Design, 37-52.**
Flemming, U.: **Rule-based Systems in Computer-Aided Architectural Design, 93-112.**
Akin, O.: **Expertise of the Architect, 173-196.**
Schmitt, G.: **ARCHPLAN: An Architectural Planning Front End to Engineering Design Expert Systems, 257-278.**

Wexler, J. (1984): *CAD '84*, 6th International Conference and Exhibition on Computers in Design Engineering, Brighton Metropole, Sussex, UK, 3-5 April 1984, Butterworth & Co. (Publishers) Ltd., Guildford, Surrey, UK.
Key Papers: Ishigami, A.: **Automated Drawing and Cost Estimatation Program for Platform Frame Construction: House 24, 38-47.**
Topping, B.H.V.; **Robinson, D.J.: Computer Aided Design of Timber Space Frames, 74-88.**
Frangopol, D.M.: **Interactive Reliability-based Optimum CAD of Framed Structures, 89-96.**
Bzymek, Z.M.: **Interactive Synthesis of Engineering Structures, 97-103.**
Krishnan, K.R.; Gopalakrishnan, M.: **A Comparative Study of Search Techniques for the Optimization of Plane Frames with Bending Moment Constraints, 104-111.**
Webster, C.A.G.: **CAPITOL: A Low Cost 3-D Modelling and Visualization System for Interior Designers and Architects, 224-235.**
Wallis, A.F.; Woodmark, J.R.: **Interrogating Solid Models, 236-243.**
Jacobsen, K.: **AUTOMODL: a 3D Modeller for Welded Steel Structures, 422-430.**
Gero, J.S.; Coyne, R.: **The Place of Expert Systems in Architecture, 529-546.**
Radford, A.D.; Hung, P.; Gero, J.S.: **New Rules of Thumb from Computer-aided Structural Design: Acquiring Knowledge for Expert Systems, 558-566.**

Wojtowicz, J. (1992): *Digital Folio*, School of Architecture, University of British Columbia, Vancouver.

Design Expert Systems

Brown, D.C. (1985): **Failure Handling in a Design Expert System**, *Computer-Aided Design*, 17(9): 436-442.

Dym (1991): **Representation and Problem Solving: The Foundations of Engineering Design**, *Planning and Design: Environment and Planning B*, 18(2).

Dym, C.L.; Levitt R.E. (1991): *Knowledge-based Systems in Engineering*, McGraw-Hill, New York.

Fazio, P.; Bedard, C.; Gowri, K. (1989): **Knowledge-based System Approach to Building Envelope Design**, *Computer-Aided Design*, 21(8): 519-527.

Fenves, S.J.; Norabhoompipat, T. (1978): **Potentials for Artifical Intelligence Applications in Structural Engineering Design and Detailing**, *Artifical Intelligence and Pattern Recognition in Computer Aided Design*, Latombe, Ed., North-Holland Publishing Company, Amsterdam, The Netherlands.

Furuta, H.; Tu, K-S.; Yao, J.T. (1985): **Structural Engineering Applications of Expert Systems**, *Computer-Aided Design*, 17(9): 410-419.

Hoey, D. (1987): *Intelligent Knowledge-based Architectural Designs*, M.Sc. Thesis, Dept. of Computer Science, University of Dublin , Dublin, Republic of Ireland.

Hosking, J.G.; Mugridge, W.B.; Buis, M. (1987): **FireCode: A Case Study in the Application of Expert Systems Techniques to a Design Code**, *Environment and Planning B: Planning and Design*, 14(3): 267-280.

Haller, F.; Lenart, M. (1986): **On the Geometry of Orthogonal Prefabricated Building Systems**, *Environment and Planning B: Planning and Design*, 13: 63-84.

Hart, A. (1985): **Knowledge Elicitation: Issues and Methods**, *Computer-Aided Design*, 17(9): 455-464.

Lansdown, J.; Roast, C. (1987): **The Possibilities and Problems of Knowledge-based Systems for Design**, *Environment and Planning B: Planning and Design*, 14(3): 255-266.

Maher, M.L. (1985): **HI-RISE and Beyond: Directions for Expert Systems in Design**, *Computer-Aided Design*, 17(9): 420-427.

McCullough, M.; Mitchell, W.J.; Purcell, P. (1990): *The Electric Design Studio: Architectural Knowledge and Media in the Computer Era*, The M.I.T. Press, Cambridge, Mass.

Key Papers:
- Stiny, G.: **What Designers Do that Computers Should, 17-30.**
- Flemming, U.: **Syntactic Structures in Architecture: Teaching Composition with . Computer Assistance, 31-48.**
- Oxman, R.; Oxman, R.: **The Computability of Architectural Knowledge, 171-185.**
- de Vries, M.; Wagter, H.: **A CADD Model for Use in Early Design Phases, 215-228.**
- Vanier, D.J.: **Hypertext: A Computer Tool to Assist Building Design, 283-300.**
- Tan, M.: **Saying What It *Is* by What It Is *Like* - Describing Shapes using Line Relationships, 201-214.**
- Oxman, R.: **Architectural Knowledge Structures as "Design Shells": A Knowledge-Based View of Design and CAAD Education, 187-199.**
- Coyne, R.D.: **Tools for Exploring Associative Reasoning in Design, 91-106.**
- Schmitt, G.: **Classes of Design - Classes of Tools, 77-90.**

Ohsuga, S. (1989): **Toward Intelligent CAD Systems**, *Computer-Aided Design*, 21(5): 315-337.

Radford, A.D.; Gero, J.S. (1985): **Towards Generative Expert Systems for Architectural Detailing**, *Computer-Aided Design*, 17(9): 428-435.

Rehak, D.R.; Howard, H.C. (1985): **Interfacing Expert Systems with Design Databases in Integrated CAD Systems,** *Computer-Aided Design*, 17(9): 443-454.

Rosenman, M.A.; Gero, J.S. (1985): **Design Codes as Expert Systems,** *Computer-Aided Design*, 17(9): 399-409.

Rosenman, M.A.; Gero, J.S.; Hutchinson, P.J.; Oxman, R. (1986): **Expert Systems Applications in Computer-Aided Design,** *Computer-Aided Design*, 18(10): 546-551.

Ruffle, S. (1986): **Architectural Design Exposed: From Computer-Aided Drawing to Computer-Aided Design,** *Environment and Planning B: Planning and Design*, 13(4): 385-389.

Shah, J.J.; Rogers, M.T. (1988): **Expert Form Feature Modelling Shell,** *Computer-Aided Design*, 20(9): 515-524.

Sham, S.H.R. (1993): **Nonmonotonic Reasoning in Design,** *ASCE Journal of Computing in Civil Engineering*, 7(1): 36-53.

Smithers, T. (1989): **AI-based Design versus Geometry-based Design or Why Design Cannot be Supported by Geometry Alone,** *Computer-Aided Design*, 21(3): 141-150.

Taylor, R.J. (1992): *TimberCon Pilot Project Demonstration of a Timber Connection Design Expert System, (runtime computer software)*, Dept. of Civil Engineering, University of British Columbia, Vancouver.

Taylor, W. (1988): *What Every Engineer Should Know About Artificial Intelligence*, The M.I.T. Press, Cambridge, Mass.

Thomson, J.V.; Marksjo, B.S.; Sharpe, R. (1987): **A Knowledge-representation Language for Engineering Design Codes (TIMBER EXPERT),** *Environment and Planning B: Planning and Design*, 14(3): 293-304.

Warman, E.A. (1978): **Computer Aided Design: An Intersection of Ideas,** *Artifical Intelligence and Pattern Recognition in Computer Aided Design*, Latombe, Ed., North-Holland Publishing Company, Amsterdam, The Netherlands.

Wood / Timber Architecture - Engineering Design

British Columbia Lumber Manufacturers Association (1960): *Post and Beam*, booklet prepared by BCLMA with University of British Columbia School of Architecture, Vancouver.

Carmichael, E.N. (1984): *Timber Engineering*, E. & F.N. Spon Ltd., London.

Germany, L. (1991): *Harwell Hamilton Harris*, The University of Texas Press, Austin, Texas.

Hansen, H.J. (1971): *Architecture in Wood (A History of Wood Building and Its Techniques in Europe and North America)*, Faber and Faber Limited, London, UK.

Hilson, B.O.; Whale, L.R.J. (1990): **Developments in the Design of Timber Joints,** *The Structural Engineer*, 68(8): 148-150.

Jacoby, H.S. (1909): *Structural Details or Elements of Design in Timber Framing*, John Wiley & Sons, New York.

Lyman, V.F. (1959): *A Survey of Residential Post and Beam Construction in Greater Vancouver 1957-1958*, DBR Technical Paper No. 70, NRC 5219, Division of Building Research, National Research Council of Canada..

Monies, F. (1961): *Wood in Architecture*, F.W. Dodge Corporation / McGraw Hill, Denmark.

Stiemer, S.F.; Prion, H.G.L. (1992): *Basic Structural Steel Design - An Electronic Textbook*, MapleSoft Development Corporation, Vancouver, BC.

Weaver, W.; Gere, J.M. (1980): *Matrix Analysis of Framed Structures*, 2nd Ed., D. Van Nostrand Co., New York.

Aesthetics

Holgate, A. (1992): *Aesthetics of Built Form*, Oxford University Press, New York.

Minai, A.T. (1989): *Design as Aesthetic Communication: Structuring Random-Order; Deconstruction of Formal Rationality*, American University Study Series XX, Fine Arts, Vol. 9, Peter Lang, Frankfurt, Germany.

Stiny, G.; Gips, J. (1978): *Algorithmic Aesthetics: Computer Models for Criticism and Design in the Arts*, University of California Press, Berkeley, Los Angeles, California.

Architectonics

Benedikt, M. ( ): **Realness and Realism: A New Direction,** *Towards an Architecture of Reality.*

Head, K.R. (1988): *In Search of 'Real' Places*, B. Arch. Thesis, School of Architecture, University of British Columbia, Vancouver.

Rykwert, J. (1982): **Semper and the Conception of Style,** *The Necessity of Artiface*, Rizzoli International Publications Inc., New York.

Seklar, E.F. (1965): **Structure, Construction, Tectonics,** *Structure in Art and Science*, Kebes, G., ed., George Braziller, New York.

Zambonini, G. (1988): **Notes for a Theory of Making in a Time of Necessity,** *Perspecta: The Yale Architectural Journal*, Rizzoli International Publications Inc., New York, 24: 3-23.

Appendix A PARTS AND SURFACES DATABASES

This Appendix contains the complete contents of the parts and surfaces database as used in this study. These databases are simple ASCII text files that can be added to without limitation by the user according to the classification procedure mentioned in the text. Figure A.1 represents the parts database, while Figure A.2 represents the corresponding part surfaces database.

PartsData HeavyTimber

Part	Description	Class	Contact	Material	Cost	Availibility	EaseofUse	Visibility	SurfaceClass	Fire
C1	ColumnContinuousAtSupport	Columns	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
C2	ColumnEndingAtSupport	Columns	End	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
R1	RafterContinuousAtSupport	Rafters	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
R2	RafterEndingAtSupport	Rafters	End	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
R3	SplicedArch	Rafters	End	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
R4	SpacedArch	Rafters	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
P1	PurlinContinuousAtSupport	Purlins	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
P2	PurlinEndingAtSupport	Purlins	End	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
B1	BeamContinuousAtSupport	Beams	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
B3	BeamSplicedAtSupport	Beams	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
B2	BeamEndingAtSupport	Beams	End	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
B5	Beam2EndsAtSupport	Beams	End	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
B4	BeamEndCantileveredOverSupport	Beams	Side	Specified	0.00	OK	OK	HiddenExposed	Unknown	Unknown
A1	AngleSingleClip	Angles	EndSideEnd	Steel	1.95	OK	OK	Exposed	SurfAnglesSS	Exposed
A2	AngleSingleBent	Angles	EndSideEnd	Steel	1.97	OK	OK	Exposed	SurfAnglesSS	Exposed
A3	AngleWeldedSingleClip	Angles	EndSideEnd	Steel	2.35	OK	OK	Exposed	SurfAnglesSS	Exposed
A4	AnglePairClipHung	Angles	SideSide	Steel	4.90	OK	OK	Exposed	SurfAngles	Exposed
A5	AnglePairClipSide	Angles	EndSideEnd	Steel	3.95	OK	OK	Exposed	SurfAngles	Exposed
A7	AnglePairBentSide	Angles	EndSideEnd	Steel	4.17	OK	OK	Exposed	SurfAnglesSP	Exposed
A8	AngleWeldedPairClipHung	Angles	SideSide	Steel	4.70	OK	OK	Exposed	SurfAngles	Exposed
A9	AngleWeldedPairClipSide	Angles	EndSideEnd	Steel	4.54	OK	OK	Exposed	SurfAnglesSP	Exposed
H1	HangerBackToBackWeldedU	Hangers	SideSide	Steel	6.65	OK	OK	Exposed	SurfHangers3	Exposed
H2	HangerWeldedLSidePlates	Hangers	EndSide	Steel	5.45	OK	OK	Exposed	SurfHangers3	Exposed
H3	HangerWeldedCentreGussetSeat	Hangers	EndSide	Steel	6.15	OK	OK	Hidden	SurfHangers3	Buried
H4	HangerWeldedCentreGusset	Hangers	EndSide	Steel	5.50	OK	OK	EmbeddedHidden	SurfHangers2	Buried
H5	HangerWeldedT	Hangers	EndSide	Steel	4.95	OK	OK	Embedded	SurfHangers3E	Buried
P1	PlateSingleNarrowSide	Plates	SideEnd	Steel	1.25	OK	OK	Exposed	SurfPlates	Exposed
P2	PlateSingleT	Plates	SideEnd	Steel	2.53	OK	OK	Exposed	SurfPlates	Exposed
P3	PlateSingleBearing	Plates	SideEnd	Steel	1.52	OK	OK	EmbeddedHiddenExposed	SurfBearing	ExposedBuried
P4	PlateSingleConcealed	Plates	EndSideEnd	Steel	3.59	OK	OK	Embedded	SurfPlatesH	Buried
P5	PlatePairNarrowSide	Plates	SideEndSide	Steel	2.50	OK	OK	Exposed	SurfPlates	Exposed
P6	PlatePairT	Plates	SideEndSide	Steel	5.06	OK	OK	Exposed	SurfPlates	Exposed
P7	PlatePairNarrowSideE	Plates	EndEnd	Steel	2.50	OK	OK	Exposed	SurfPlates	Exposed
P8	PlateSingleConcealedE	Plates	EndEnd	Steel	3.59	OK	OK	EmbeddedHidden	SurfPlatesH	Buried
D0	None	WWDistributor	SideSide	Unknown	0.00	OK	OK	EmbeddedPartlyFully	Unknown	ExposedBuried
D1	Dowel	Dowels	SideEnd	Steel	1.33	OK	OK	PartlyFully	Unknown	Exposed
D2	BoltWithNut	Bolts	SideSide	Steel	3.03	OK	OK	EmbeddedPartlyFully	Unknown	ExposedBuried
D3	WasherPair	Washers	SideSide	Steel	0.88	OK	OK	EmbeddedPartlyFully	Unknown	ExposedBuried
D4	LagScrew	Lagscrews	SideSide	Steel	0.29	OK	OK	PartlyFully	Unknown	ExposedBuried
D5	GlulamRivet	Rivets	SideSide	Steel	0.53	OK	OK	Fully	Unknown	ExposedBuried
D6	Nail	Nails	SideSide	Steel	0.25	OK	OK	EmbeddedPartly	Unknown	ExposedBuried

Figure A.1 Parts Database

D7	WoodScrew	Screws	SideSide	Steel	0.08	OK	OK	EmbeddedPartly	Unknown	ExposedBuried
D8	LaggedShearPlate	ShearPlateLS	SideSide	Steel	4.94	OK	OK	EmbeddedPartlyFully	Unknown	ExposedBuried
D9	BoltedShearPlate	ShearPlateB	SideSide	Steel	4.94	OK	OK	EmbeddedPartlyFully	Unknown	ExposedBuried
D10	Burieddowel	Dowels	EndEnd	Steel	1.33	OK	OK	Embedded	Unknown	Buried
S1	SplitRingSS	SplitRings	SideSide	Steel	3.61	OK	OK	Embedded	SurfRingsFP	ExposedBuried
S2	SplitRingSE	SplitRings	SideEnd	Steel	3.61	OK	OK	Embedded	SurfRingsPP	ExposedBuried
S3	SplitRingES	SplitRings	EndSide	Steel	3.61	OK	OK	Embedded	SurfRingsPP	ExposedBuried
S4	SplitRingEE	SplitRings	EndEnd	Steel	3.61	OK	OK	Embedded	SurfRingsE	ExposedBuried
S5	ShearPlatesSS	ShearPlates	SideSide	Steel	4.94	OK	OK	Embedded	SurfShearPlatesFP	ExposedBuried
S6	ShearPlatesSE	ShearPlates	SideEnd	Steel	4.94	OK	OK	Embedded	SurfShearPlatesPP	ExposedBuried
S7	ShearPlatesES	ShearPlates	EndSide	Steel	4.94	OK	OK	Embedded	SurfShearPlatesPP	ExposedBuried
S8	ShearPlatesEE	ShearPlates	EndEnd	Steel	4.94	OK	OK	Embedded	SurfShearPlatesE	ExposedBuried
HP1	Pipe	Pipes	SideEnd	Steel	5.50	OK	OK	EmbeddedHiddenExposed	SurfRods	Buried
D11	Adhesive	Adhesives	SideSide	Specified	3.53	OK	OK	Embedded	Unknown	Buried
TJ1	MortiseTenon	WWTimberJoinery	EndSide	Wood	17.50	OK	OK	EmbeddedHidden	SurfJoinery	Buried
TJ2	DoveTail	WTimberJoinery	EndSide	Wood	15.53	OK	OK	EmbeddedHidden	SurfJoinery	Buried
TJ3	BeveledShoulder	WWTimberJoinery	EndSide	Wood	12.95	OK	OK	EmbeddedHidden	SurfJoinery	Buried
TJ4	Birdsmouth	WWTimberJoinery	SideSide	Wood	13.52	OK	OK	EmbeddedHidden	SurfJoinery	Buried
TJ5	HousedMortiseTenon	WWTimberJoinery	EndSide	Wood	19.95	OK	OK	EmbeddedHidden	SurfJoinery	Buried
N1	LapFastenersOnly	WWNoTransferer	SideSide	None	0.00	OK	OK	Embedded	SurfNoneSideSide	ExposedBuried
N2	ButtFastenersOnly	WWNoTransferer	EndEnd	None	0.00	OK	OK	Embedded	SurfNoneEndEnd	ExposedBuried
N3	EndFastenersOnly	WWNoTransferer	EndSide	None	0.00	OK	OK	Embedded	SurfNoneEndSide	ExposedBuried
N4	SideFastenerOnly	WWNoTransferer	SideEnd	None	0.00	OK	OK	Embedded	SurfNoneSideEnd	ExposedBuried

Figure A.1 *continued*

SurfaceData HeavyTimberParts

PartClass	Name	Owner	LocationPlane	MovementFastened	MovementUnfastened	MatingSurfaces	PossibleDistributors
SurfAngles	a	Taker	XY	xyzXYZ	z	bc	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfAngles	b	Taker	XY	xyzXYZ	z	ad	None
SurfAngles	c	Giver	YZ	xyzXYZ	x	ad	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfAngles	d	Giver	YZ	xyzXYZ	x	bc	None
SurfAnglesSP	ab	Taker	XY	xyzXYZR	z	cd	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfAnglesSP	cd	Giver	YZ	xyzXYZR	x	ab	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfAnglesSS	ab	Taker	XY	xyzXYZ	z	cd	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfAnglesSS	cd	Giver	YZ	xyzXYZ	x	ab	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfHangers3	a	Taker	XY	xyzXYZR	z	bc	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfHangers3	b	Giver	YZ	xyzXYZR	xXR	ac	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBDowelsAdhesivesNone
SurfHangers3	c	Giver	XZ	xyzXYZR	Y	ab	NailsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfHangers3E	a	Taker	XY	xyzXYZR	z	bc	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfHangers3E	b	Giver	YZ	xyzXYZR	xXR	ac	NailsBoltsLagscrewsDowelsAdhesivesNone
SurfHangers3E	c	Giver	XZ	xyzXYZR	Y	ab	NailsLagscrewsAdhesivesNone
SurfHangers2	a	Taker	XY	xyzXYZR	z	b	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfHangers2	b	Giver	YZ	xyzXYZR	xXR	a	NailsBoltsLagscrewsDowelsAdhesivesNone
SurfPlates	a	Taker	YZ	xyzXYZR	xXR	b	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfPlates	b	Giver	YZ	xyzXYZR	xXR	a	NailsBoltsLagscrewsRivetsShearPlateLSShearPlateBAdhesivesNone
SurfPlatesH	a	Taker	YZ	xyzXYZR	xXR	b	NailsBoltsLagscrewsDowelsAdhesivesNone
SurfPlatesH	b	Giver	YZ	xyzXYZR	xXR	a	NailsBoltsLagscrewsDowelsAdhesivesNone
SurfBearing	a	Taker	YZ	xyzXYZR	xXR	b	AdhesivesDowelsNone
SurfBearing	b	Giver	YZ	xyzXYZR	xXR	a	AdhesivesDowelsNone
SurfRods	a	Taker	YZ	xyzXYZR	xXR	b	BoltsDowelsAdhesivesNone
SurfRods	b	Giver	YZ	xyzXYZR	xXR	a	BoltsDowelsAdhesivesNone
SurfRingsFP	a	Taker	YZ	xyzXYZR	xyXY	b	BoltsLagscrews
SurfRingsFP	b	Giver	YZ	xyzXYZR	xyXY	a	BoltsLagscrews
SurfRingsPP	a	Taker	YZ	xyzXYZR	xyXY	b	Lagscrews
SurfRingsPP	b	Giver	YZ	xyzXYZR	xyXY	a	Lagscrews
SurfRingsE	a	Taker	YZ	xyzXYZR	xyXY	b	None
SurfRingsE	b	Giver	YZ	xyzXYZR	xyXY	a	None
SurfShearPlatesFP	a	Taker	YZ	xyzXYZR	xyXY	b	BoltsLagscrews
SurfShearPlatesFP	b	Giver	YZ	xyzXYZR	xyXY	a	BoltsLagscrews
SurfShearPlatesPP	a	Taker	YZ	xyzXYZR	xyXY	b	Lagscrews
SurfShearPlatesPP	b	Giver	YZ	xyzXYZR	xyXY	a	Lagscrews
SurfShearPlatesE	a	Taker	YZ	xyzXYZR	xyXY	b	Burieddowel
SurfShearPlatesE	b	Giver	YZ	xyzXYZR	xyXY	a	Burieddowel
SurfJoinery	a	Giver	XY	xyzXYZR	xyXY	Unknown	BurieddowelDowelsAdhesivesNone
SurfJoinery	b	Taker	XY	xyzXYZR	xyXY	Unknown	BurieddowelDowelsAdhesivesNone
SurfNoneEndEnd	a	Giver	XY	xyzXYZR	xyXY	Unknown	BurieddowelAdhesivesNone
SurfNoneEndEnd	b	Taker	XY	xyzXYZR	xyXY	Unknown	BurieddowelAdhesivesNone

Figure A.2 Surfaces Database

SurfNoneSideSide	a	Giver	YZ	xyzXYZR	yzYZ	Unknown	NailsBoltsLagscrewsDowelsAdhesivesNone
SurfNoneSideSide	b	Taker	YZ	xyzXYZR	yzYZ	Unknown	NailsBoltsLagscrewsDowelsAdhesivesNone
SurfNoneEndSide	a	Giver	XY	xyzXYZR	xyXY	Unknown	NailsBoltsLagscrewsDowelsAdhesivesNone
SurfNoneEndSide	b	Taker	YZ	xyzXYZR	yzYZ	Unknown	NailsBoltsLagscrewsDowelsAdhesivesNone
SurfNoneSideEnd	a	Giver	YZ	xyzXYZR	yzYZ	Unknown	NailsBoltsLagscrewsDowelsAdhesivesNone
SurfNoneSideEnd	b	Taker	XY	xyzXYZR	xyXY	Unknown	NailsBoltsLagscrewsDowelsAdhesivesNone

Figure A.2 *continued*