

**NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:**

The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

**Representing and Recording Design Intent:  
A Progress Report**

R. Ganeshan, S. Finger, J. Garrett

EDRC 12-52-92

# Representing and Recording Design Intent : A Progress Report

Rajaram Ganeshan<sup>1</sup>, Susan Finger<sup>2</sup> and James Garrett<sup>2</sup>

Department of Civil Engineering

Carnegie Mellon University

## Abstract

This research outlines a design methodology for recording the intent behind decisions in a design process. The design problem is described as a conjunction of *design objectives*, which are high level descriptions of the requirements that the design artifact must meet (functional, financial, constructability, maintainability, disposability etc.). We use a combination of informal text, first-order predicate logic and arithmetic constraints to express design objectives. Importance can be associated with objectives. In this design process, the objectives can be addressed in some order by *focussing* on a subset of them. Objectives are achieved by a process of *refinement* which may result in a set of alternative bindings for the design variables involved, or, in alternative decompositions for the objectives. Assumptions may be made in the generation of the alternatives. The alternatives are *evaluated with* respect to other objectives and one alternative is *selected*. The design is completed when the current set of objectives is satisfied. Thus, the designer does not manipulate the form of the artifact itself; rather, the designer manipulates the objectives and selects alternatives, thereby revealing the intent behind the decisions.

We also describe a representation for the record of this design process. The record can be used to identify the objectives and assumptions responsible for design decisions and for identifying the objectives and assumptions affected when decisions are modified. We are developing an interactive computer-based environment in the domain of residential and small office buildings to demonstrate this approach.

---

<sup>1</sup> Doctoral Candidate

<sup>2</sup> Assistant Professor

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>                                    | <b>2</b>  |
| 1.1      | Motivation . . . . .                                   | 2         |
| 1.2      | Research Goals. . . . .                                | 5         |
| <b>2</b> | <b>Background</b>                                      | <b>5</b>  |
| 2.1      | Computable Representations. . . . .                    | 5         |
| 2.2      | Semi-computable - Hypermedia based approaches. . . . . | 6         |
| <b>3</b> | <b>Overview of Approach</b>                            | <b>8</b>  |
| 3.1      | Overview of the Design Process Model. . . . .          | 8         |
| 3.2      | Design Intent . . . . .                                | 10        |
| 3.2.1    | Using Design Intent . . . . .                          | 11        |
| 3.3      | Characteristics of the approach. . . . .               | 12        |
| 3.4      | Issues related to design objectives. . . . .           | 14        |
| 3.4.1    | Objective Definitions. . . . .                         | 14        |
| 3.4.2    | Relative Importance . . . . .                          | 16        |
| 3.4.3    | Satisfaction . . . . .                                 | 16        |
| 3.5      | Issues for a computer-based implementation. . . . .    | 17        |
| <b>4</b> | <b>Computable Representation of Design Intent</b>      | <b>18</b> |
| 4.1      | Representation of Design Data . . . . .                | 18        |
| 4.2      | Design Objectives, Importance and Assumptions. . . . . | 20        |
| 4.3      | The Design Process. . . . .                            | 26        |
| 4.3.1    | Focus. . . . .   | 26        |
| 4.3.2    | Refine. . . . .  | 26        |
| 4.3.3    | Evaluate. . . . .                                      | 31        |
| 4.3.4    | Select . . . . .                                       | 32        |
| 4.3.5    | Resolve . . . . .                                      | 35        |
| <b>5</b> | <b>Recording the history of the Design Process</b>     | <b>30</b> |
| 5.1      | The Design State. . . . .                              | 30        |
| 5.2      | The Activity Record. . . . .                           | 31        |
| 5.3      | The Organization of the Design Record. . . . .         | 32        |
| 5.3.1    | Objectives. . . . .                                    | 32        |
| 5.3.2    | Assumptions. . . . .                                   | 33        |
| 5.3.3    | Variables: Design and Exogenous. . . . .               | 34        |
| 5.3.4    | Alternatives. . . . .                                  | 35        |
| 5.3.5    | Design Operators . . . . .                             | 36        |
| 5.3.6    | The Design Record. . . . .                             | 37        |
| <b>6</b> | <b>Example</b>   | <b>47</b> |
| <b>7</b> | <b>Summary</b>   | <b>58</b> |

# 1 Introduction

Civil engineering facilities are large, one-of-a-kind, last for many years, and require that many diverse perspectives participate in their design. The duration of the building delivery process itself is often measured in years. During this period many decisions are made, affecting many parts of the design. The motivations for these decisions are usually recorded informally in the designer's notebook, if they are recorded at all. When changes to the design must be made, it is often difficult to reconstruct why certain decisions were made and to predict the consequences of proposed changes.

A design may be altered for many reasons. During the design process, previous decisions may need to be altered or retracted because of changes in specifications or assumptions. Events during the construction of a facility may require design changes. Renovations involve making design changes to completed facilities. A new design may be generated by modifying the design for an existing structure. To properly make changes in a design, the intent behind the original decision must be known. Changes must either be consistent with the original intent, or the original intent must be explicitly modified.

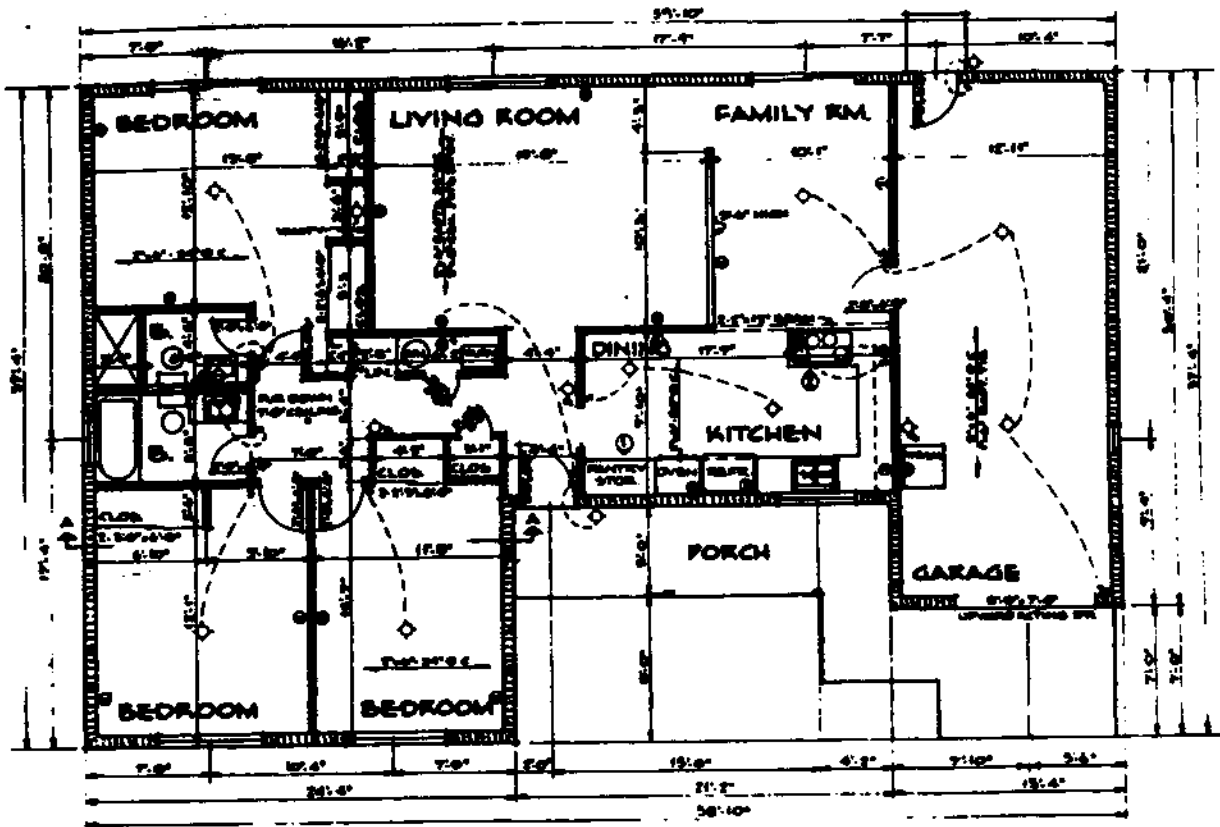
## 1.1 Motivation

Consider the floor-plan of a building shown in Figure 1. It is one of a set of final design documents prepared at the end of the building design process. Suppose one of the windows needs to be resized or relocated for one of the reasons indicated earlier. The following questions need to be answered:

- What are the ramifications of the change? For example, the area of glazing will be different.
- How does it affect the designer's original intent? For example, a reduction in size might reduce the amount of daylight required by a task within the building space.
- What other changes have to be made so that the original intent is maintained? For example, increasing the size of some other window or perhaps providing more artificial lighting etc.

Most of these questions cannot be answered with the information recorded in the drawings. We are specifically interested in the answer to the second question: why was the particular size of the window chosen? It may have been chosen to meet some daylight requirement. The designers responsible for the specifications may or may not be able to recollect the reason behind a particular decision. We refer to the reasons behind design decisions as *design intent*. The following paragraphs introduce our approach to this problem.

We view design as a process of refinement of objectives. See Figure 2. Objectives are requirements at different levels of abstraction that drive the design process. Examples of highly abstract objectives are: 1) functionality, (2) aesthetics, (3) economy, (4) constructability or manufacturability, (5) maintainability (the artifact must be maintained during its service life), and (6) disposability (the artifact must be disposed of at the end of its service life). Another class of design objectives is concerned with planning the design process. For example, process objectives might be to minimize the time to complete a design and to minimize the number of design iterations. These objectives guide the decision process during design. Objectives may have different levels of importance. This information is used to make tradeoffs when there is a conflict among the objectives.



PLOORJ\* PLAN SCALE 1/8" = 1'-0"

Figure 1: Example Floor plan

The design begins with a set of objectives that represent the highest level specification (level  $n$  in Figure 2). Objectives are refined into other objectives and/or into design alternatives. A design decision is made by choosing from among these alternatives so that a certain group of objectives are satisfied. At any stage in the process, the design is represented by the decisions that have been made so far. The refinement, of an objective may be thought of as the process of meeting the objective. Thus an objective may be achieved by decomposing it into a set of new objectives, or by making a design decision, or by both making a decision and creating a new set of objectives. This process of refinement continues until all current objectives in the design are met.

Given this brief definition of the design process, we can now define the intent for a decision,  $D$ , as follows. Let  $S_0$  be the set of objectives that were directly responsible for  $D$ .  $S_0$  includes the objectives that were used to generate the alternatives for the decision  $D$ , and those objectives that were used to choose from among the alternatives. The set  $S_0$  is a subset of the objectives in level  $n$  (Figure 2). For each  $O_i$  in  $S_0$ , let  $s_{0i}$  be the set of objectives that were directly involved in the refinement that created  $O_i$ . Define  $S^0$  to be the union of all sets  $s_{0i}$ . i.e.,  $S^0 = \bigcup_{O_i \in S_0} s_{0i}$ . Similarly for each objective  $O_i$  in  $S_1$ , let  $s_{1i}$  be the set of objectives that were involved in the refinement that

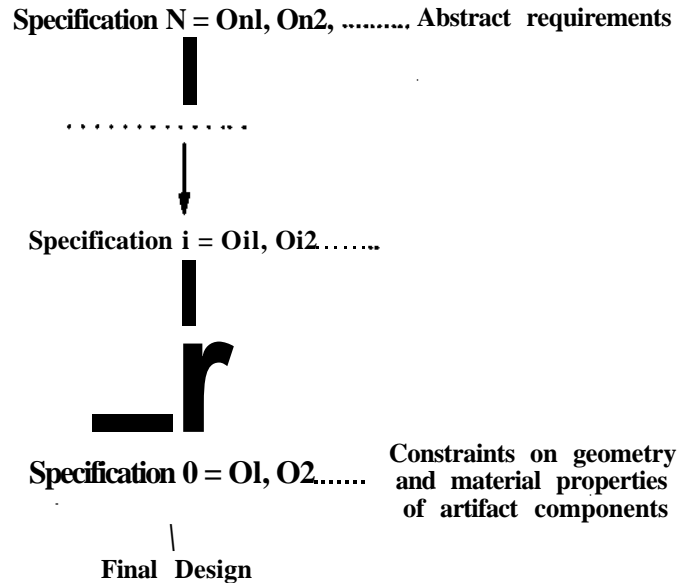


Figure 2: Design as Refinement of Objectives

created  $O_i$ . Define  $S_0$  to be the union of all sets  $s_u$ . The intent for decision  $D$  is the union from level 0 to  $n$ , that is,  $\text{intent}(D) = S_0 \cup \bigcup_{i=1}^n S_i$ .

Our approach to capturing design intent is to maintain a record of the design process starting from the highest-level objectives, including all their refinements, leading to the decision. Such a documentation of the design process is useful for:

- Explanation - to explain how and why a particular decision was made;
- Verification - to determine if characteristics of the final design are consistent with the intended characteristics;
- **Modification** - to predict the effect of making changes to the design: and.
- **Re-use** - to synthesize a design from a previous design with a similar specification.

**In summary, there** is a need to document more information about the design than is currently done by means of design drawings and specifications. Some information about the design process may be recorded informally in the designer's notes. However, such notes are unstructured and often incomplete and hence, are not useful. In this approach we attempt to develop a model that identifies the information used to make design decisions reflected in the drawings and specifications.

## 1.2 Research Goals

The goals of this research are to:

- identify the components of design intent, i.e., the information used to make design decisions: and
- develop computable representation(s) for these components such that the record of the design process can be used for: verification, explanation, modification and reuse in an automated fashion.

Recently, others have attempted to address this problem or some related problem. We will review these approaches in the following section.

## 2 Background

This section reviews some of the previous and current research concerning the recording of design intent. This research work may be divided into two categories: 1) computable representations for the record of intent, and 2) non-computable (or semi-computable) representations for the record of intent. By computable we mean that it is possible to interpret and use the information in the record in an automated fashion. The difference will become clear as the approaches are discussed in more detail. We begin with the computable approaches.

### 2.1 Computable Representations

Much of the research in recording design intent has been done for software design. We will discuss two efforts: 1) XPlain [26], in which the motivation was to be able to explain the behaviour of expert consulting programs, and 2) Popart [29], in which the main motivation was to make it easier to maintain large programs and to reuse existing software. Some of the earliest work in capturing design intent was in the development of the XPlain system [26]. The XPlain system presents an approach for capturing the reasoning that goes into the design of an expert system. The goal is to generate explanations of the system's behavior. An automatic programmer is used to generate the expert performance program. As the program is generated, a refinement structure is created which gives the explanation routines access to the decisions made during the creation of the program. This refinement structure can be thought of as a trace of the process that created the expert system.

Modification is particularly important in the maintenance of complex software. Wile developed a language that allows the user to express the design decisions made in the development of a program [29]. The language, called Paddle, has definition facilities for transformations, strategies, and plans capable of expressing different types of goal structures. These strategies such as divide-and-conquer are expressed using English-like descriptions in Paddle. The history of the development of a program from specification to final implementation is captured in a development structure that can be replayed to determine the effect of changes to the original specification. The language does not include justifications for the decisions made during the development of a program.

An important motivation for capturing design intent has been the notion of redesign in which a new design is created by modifying a previous design solution. Redesign is the uiMkrlyinu.



methodology used in design by derivational analogy [16] and in case-based reasoning [11]. Redesign, a system for circuit design, uses the solution to a previous design to obtain a new design for a similar set of specifications [14]. It uses a representation called a *Design Plan* to capture the purpose for each module of the circuit. The design plan describes how the specifications were decomposed to develop a circuit (that is, the rules that were fired) but does not capture why a particular decomposition was used.

In the areas of civil or mechanical engineering design, little effort goes into capturing the design intent. The only records of such information are the informal notes in a designer's notebook. We describe two recent efforts that begin to address this problem. Rossignac et al. present an approach in which the designer interactively designs a geometric model of a mechanical part. The commands are captured as unevaluated specifications of operations [21]. The system provides tools for editing, parameterizing, and combining these sequences so they can be applied to a family of models. The sequence of operations explains how the final model was determined but does not answer why the particular set of operations were used.

Thompson and Lu describe a design environment (AIDEMS - An Intelligent Design Evolution Management System) that allows designers to express design strategies and tactics used in developing the final product description by successively refining the specification (initial product description) [27]. The system uses artificial intelligence techniques for resource scheduling, explanation of design assertions, execution of design revisions, inconsistency detection and exploration of design alternatives. In the AIDEMS design methodology all activities are classified as either strategic or tactical. Strategic activities involve the development of plans and tactical activities involve the execution of plans resulting in the product description. The activities executed during the design process are recorded in documents that include the activity's preconditions, a description of the executed activity, and a listing of the resulting assertions. The design product is described in terms of assemblies and components. Design specifications are expressed in mathematic, algorithmic, temporal, and textual formats. However, in this approach there is no representation for tradeoffs between conflicting specifications.

Another ongoing effort in this direction is the work of Craig Howard and his students at Stanford [6], who looking at problem of capturing project specific knowledge in the context of facility life-cycle. Project specific knowledge is defined as the knowledge specific to a particular project and its supporting rationale, including design decisions that link different ingredients of data and knowledge. They are investigating the character and form of this knowledge through case studies. They propose an object-oriented model as the general scheme for representing this knowledge.

## 2.2 Semi-computable - Hypermedia based approaches

The non-computable or semi-computable efforts are based on a view of design as negotiation, sometimes referred to as the issue-based deliberation model. The roots of this work lies in the IBIS method [20] for policy decision-making, that addresses the design process for complex problems as a process of negotiation among different groups (eg. the client, contractor, architect etc.) with different stakes in the problem. In the IBIS model the design problem is stated in the form of *issues*. An issue can have many *positions* each of which responds to the issue. Each position may have *more arguments* supporting or speaking against it. Issues can suggest other issues, or be *superjunctive*,

or generalized into other issues. gIBIS [3], (Figure 3) is a graphical tool that supports the IBIS model

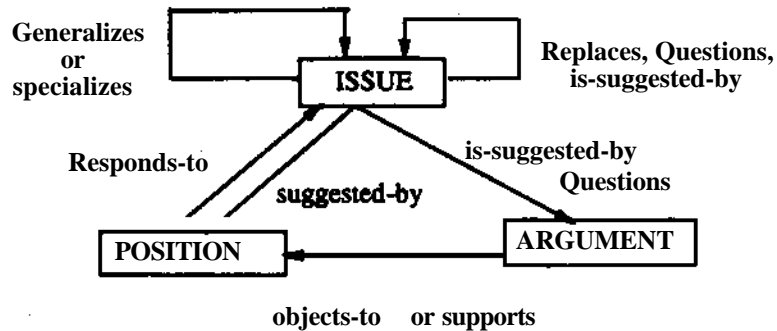


Figure 3: Issue based deliberation model(after Conklin, Begeman, 1988)

in a hypertext like environment. Another related work is the Potts and Bruns model [18], that is also influenced by the IBIS model. The model treats the design process as a network consisting of deliberation and artifact nodes. These efforts take a more realistic view of the design process. They provide a means of organizing the deliberations in the design process within the computer. Since they use informal representations for the contents of the various nodes in the model, they must rely on the designer to identify the effects of a design change, i.e., the designer must browse through the network to identify possible ramifications of changing some design decision.

The PHI (Procedural Hierarchy of Issues) model [13] extends and modifies the IBIS model. In the PHI model issues are related by a single relationship: the serve relationship. Issue A serves issue B if resolving A helps resolve B. Two kinds of serve relationships are identified between issues. 1) Subissue and 2) Antecedent. The difference between the two relationships is temporal, i.e.. A subissue of B implies that A is raised after B; A antecedent B implies that B is raised after A. A single issue is distinguished as a prime issue. The resolution of this issue terminates the entire process. All other issues serve the prime issue. McCall argues that, subissue relationships are usually greater than antecedent relationships. Also, antecedent relationships occur between subissues of the same issue. Hence, omitting these antecedent relationships yields a tree-like form for the issue relationships as shown in Figure 4. The nature of these relationships is described by McCall as being quasi-hierarchical because the same issue might be a sub-issue to two issues and sometimes there may be cycles. Like in the IBIS approach, the resolution of an issue is carried out by proposing answer\* and argument\* for the answers (no distinction is made between an argument for or against). It is possible to have arguments on other arguments thus forming argumentative hierarchies. Answer may be given at different levels of generality or specificity thus generating answer hierarchies.

Like gIBIS, PHI also uses informal textual representations. The argument for the informal representation is that such systems address the early stages of the design problem where formal representations are difficult to find. Informal representations will depend on human interpretation while formal representations will enable automation of processes such as backtracking and determining ramifications of a modification.

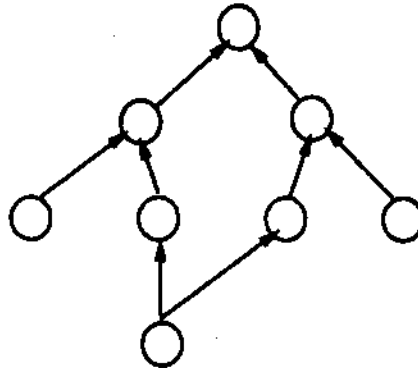


Figure 4: Issue (Quasi)Hierarchy (after McCall, 1990)

### 3 Overview of Approach

This section provides an overview of our approach to the capturing design intent. The first section provides an overview of the proposed design process model. The following section describes the definition of design intent within this model and the how it is to be used. The third section discusses some characteristics of the approach. The notion of design objectives is central to this approach. The next section discusses some issues related to objectives. The final section summarizes the issues that have to be addressed in using this methodology in a computer-based environment.

#### 3.1 Overview of the Design Process Model

The initial specification for a design artifact can be stated as a conjunction (AiND) of objectives with associated importance information. In the ideal case, if the initial statement of the problem were complete and consistent so that a unique solution (i.e artifact) is identifiable then the problem is solved. This case will never occur in practice. This thesis is concerned with the more realistic problem of identifying a design methodology whereby the designer begins with an initial set of incomplete and inconsistent objectives that maybe changed (some objectives may be deleted and new ones added or the importance associated with an objective is changed) until a satisfactory design solution (artifact) along with the objectives that define that solution are obtained. This is carried out by a design process that is characterized by the following activities:

- determining *design focus*
- *refining* an objective
- *evaluating* design alternatives with respect to objectives
- *selecting* an alternative based on the results of evaluations

- **resolving conflicts among various** design objectives

This notion of the types of activities is derived from an approach used in a system called SPEX (Standards Processing **EX**pert) [7]. **Before describing** the process in more detail the following definitions are useful:

- **Objectives** are requirements that the final design must meet such as those involving function, aesthetics, economy, constructability, maintainability and disposability. Objectives may have different levels of importance associated with them.
- Decisions constitute the descriptions of the design artifact (e.g., shape-of-element = wide-flange). They are the result of decision-making activities.
- Alternatives are sets of objectives and decisions that represent the different ways in which some of the objectives may be achieved in the design. For example, the sound isolation of a space in a building from a noise source may be achieved either by: 1) placing the space far from the source (sound intensity reduces with distance) or, 2) using an acoustic obstruction between the space and source.
- **Operators** contain knowledge as to how to achieve objective(s) in the design. For example, the knowledge about how to achieve sound isolation or how to solve simultaneous equations etc.
- Assumptions are the default requirements that are established during the course of the design process. They are made to account for the lack of information about some design entity. They are usually based on the experience and judgement of the designer. An assumption differs from an objective in that it is not strong requirement.
- Heuristics are rules that may guide the selection of a particular alternative for achieving a set of objectives. There is a certain amount of uncertainty associated with them but it is not quantified.

**Focusing.** The first step in the design process is to determine which objectives to refine. Often one objective is chosen before another due to an information dependency between objectives. For example, the dimensions of a space will be determined before the openings in the space are located and sized. Focusing may **have** implications for the efficiency of the design process, i.e., it may affect the time to arrive at a **solution**. In this approach we record the sequence in which the objectives are addressed but **do not capture the** information for why they were addressed in that sequence.

**Refining.** In our model, an *objective refinement* results in an objective being decomposed or specialized into other objectives or a decision. The subobjectives represent conditions that must exist in the **design** for the original objective to be satisfied. It is possible that there are different ways (alternatives) **to** achieve an objective(s).

**Evaluating.** At this point the feasibility of the alternatives is determined with respect to the other objectives in the design. The evaluation of the alternatives must be based on the information available at the current level of abstraction. Thus, it may not always be possible to clearly determine the feasibility. In this case, heuristics may be used to eliminate one of alternatives.

**Selecting.** The process of selecting involves choosing the alternative that best satisfies a set of objectives. The difference between selection and evaluation is that in the case of evaluation the alternative that fails the evaluation with respect to some objective is eliminated, while in the case of selection the alternative that is not selected is not necessarily eliminated. The selection process makes use of knowledge about the relative importance of the objectives. Sometimes, it may not be possible to make a clear choice at this point, and the designer might 1) use heuristics to select a promising alternative or 2) arbitrarily select any one of the alternatives.

**Resolving.** All of the alternatives may be eliminated in the process of evaluation. In other words, none of the alternatives may satisfy all the relevant objectives, so no alternatives are available for selection. This interaction among objectives may be resolved by 1) changing an arbitrary selection made at a previous step, 2) changing a heuristic evaluation or selection made at a previous step, 3) changing some assumptions. 4) eliminating or introducing an objective, and, 5) changing the importance associated with some objective. This is done during a resolution activity.

Finally, the design is completed when a level of detail sufficient to construct (or manufacture) the artifact has been reached and the current set of objectives are achieved.

## 3.2 Design Intent

We define intent to be the hierarchy of objectives that was developed during the design process for an artifact. The generation of this hierarchy is influenced by a number of factors as indicated below:

- Alternatives that were considered (this is dependent on the refinement operator)
- Assumptions that were made during the generation of an alternative
- Evaluation of the alternatives and the criteria for selection
- Trade-offs among objectives (importance may influence the choice of an alternative)
- The focus sequence, i.e., the sequence in which the objectives are addressed
- Heuristics that may have influenced the choice of a particular alternative
- Local environment, i.e., given conditions within which the objectives were achieved
- Revision of an objective (the designer may not be satisfied with a solution generated by the current set of objectives or a solution may not exist for the current set of objectives, which causes a revision of some objectives).

Thus, the final hierarchy of objectives alone is not sufficient to explain the decisions made to achieve the final design. We also need to record how this hierarchy came about, i.e., why this form was obtained for the hierarchy given the other alternative forms that could have occurred. In our approach, the history of the design process captures not only the current intent for the design in terms of the final objective hierarchy, but also how this hierarchy came about. The following section describes how this record of intent, may be used.

### 3.2.1 Using Design Intent

This section describes how the information we have captured in the design record may be used both during the design process and subsequently.

**Explanation and Modification.** One use for the record is to explain why a particular decision was made. Decisions can be related to either 1) the design process i.e. why was a particular decomposition of the objectives chosen, or 2) to the design artifact. The model of decision making in our design process is similar for both cases. The decision is motivated by a set of objectives. A subset of this set of objectives (refinement) is used to generate different alternatives for the decision. The other subset is used to evaluate the alternatives and select from among them. Some of the alternatives will be eliminated because they violate one or more objectives. For the alternatives that are remaining, the selection criteria along with importance information will indicate why one of the alternatives is superior to the others. So, the objectives that caused an alternative to be eliminated will form the argument against that alternative. The objective used in selection will explain why one of the alternatives (the one chosen) was superior to other non-eliminated alternatives. To explain a decision, we need to 1) locate it in the record, 2) identify the various alternatives considered for the decision, 3) identify the objectives used to generate the alternatives and to evaluate and select from them.

Another use for the record is to identify the ramifications of changes the design. For example, moving or resizing a window might affect the daylight factor objective. In general we can express such situations as the following question: What objective(s) is(are) violated when the following changes are made to the design? The history of the design process records the objectives that were refined to generate and evaluate alternative(s). Thus the link between decisions and objectives is established.

Reuse. There are various kinds of reuse that are possible within the proposed approach:

- Objectives: An objective has to be defined only once. Subsequently, it may be instantiated in any other design situation. Thus, the more general we make the definition of an objective, the more amenable it will be to reuse.
- Operators: The same argument holds for operators. As more objectives and operators are defined the system acquires more knowledge. This knowledge indicates how to achieve different kinds of objectives. Some of these are domain independent (e.g. mathematical programming algorithms) while others are domain specific (e.g. sound isolation). Also, this is general knowledge in the sense that it indicates all the possible ways in which an objective may be achieved. However, in a particular design situation one or more of these alternatives may not be applicable. It may be possible to use heuristics to recognize situations in which certain alternatives may not be applicable. This is discussed in the following item.
- Process: It may be possible for the user to go over the record of the solution process a posteriori and identify places where wrong decisions were made and identify heuristics (may take the form of rules) that might at the very least improve the process for the same design problem. This is a feedback that can improve the design problem solving for the class of problem being solved.

### 3.3 Characteristics of the approach

The paradigm presented here requires the designer to follow a particular design methodology that creates a model of the designer's intent. Typically, this model of intent is inconsistent in that it admits no solution or the designer is not satisfied with the solution that is produced by it. In either case the designer wishes to alter the original model of intent. This is done during both the selection and resolve stages of our model or explicitly by the designer (modifying objectives). This cycle continues until the designer is satisfied with the design solution. At this point, because of the nature of the paradigm, the objectives motivating the final design and also the evolution of these objectives over the course of the design process is recorded.

There are two distinguishing characteristics of this approach: 1) the designer manipulates the design through the objectives, and, 2) it is a systematic approach to design. Conventional CAD systems provide the designer with tools to directly manipulate the objects in the design. In these systems, the designer uses these tools to create a design that satisfies some intent. There is no representation for this intent within the system. In contrast, we provide designers with tools to express intent (in the form of objectives) and to transform that intent into a design.

The design process is often described in terms of the steps shown in Figure 5. The approach

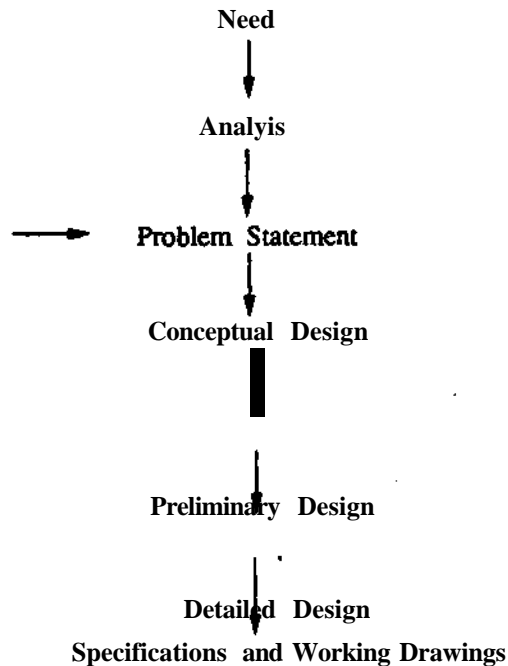


Figure 5: Steps in the Design Process

we have described begins with the statement of objectives as indicated by the arrow in Figure V A-

we have stated earlier, this set of initial objectives does not have to be complete or final or consistent. Objectives may be added and changed over the course of this process. We believe that our model is applicable over all of the stages, i.e., conceptual to detailed design.

A distinction is often made between innovative and routine design problems. In our view the distinction arises from how the objectives for these problems are stated. In routine design problems a number of implicit assumptions are made in describing objectives, whereby the space of solutions is reduced to some previously known set. If objectives are stated in a more general way such that the space of solutions is larger than the previously known set, then there is a possibility of arriving at a new solution, i.e., one that has not been seen previously.

Our design process model is similar to some prescriptive approaches found in the literature [12], [17]. While Pahl and Beitz [17] address machine design, and Lin and Stotesbury [12] address building design, both approaches involve the generation and evaluation of alternatives at every level of decision-making. Consider for example, the steps in the conceptual design process according to Pahl and Beitz shown in Figure 6.

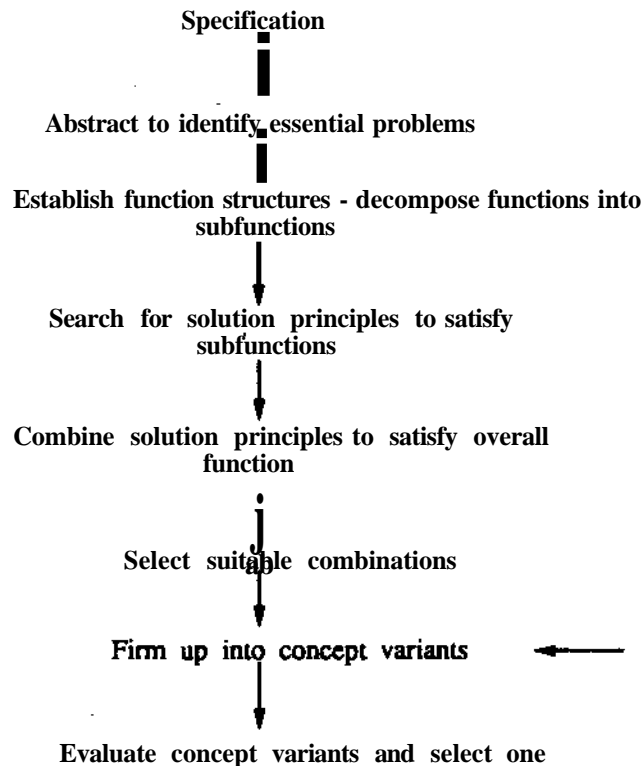


Figure 6: Conceptual Design: Pahl and Beitz



When **functional objectives** are achieved independently (if they can be achieved that way) **the design may not be an efficient solution**. This is because possibilities of interactions between objectives are ignored (**again, assuming that** it is possible to achieve them independently). For example, consider an objective **that requires** an opening for circulation between an activity space, A, and the outside, and **another objective that** requires a certain amount of daylight in the same activity space. Since both of **these objectives require** openings to the outside (although the dimensions may be different) there is a possibility **that** the same opening can be used to achieve both objectives (i.e., an opening that is both a door and allows daylight to enter the space). If these objectives are achieved independently of each other assuming that the other objectives in the design make this possible (i.e., it may be that adjacency requirements with other spaces and A leaves enough leeway for a separate door and window) then the final design will have one opening for daylight and another for circulation. In the Pahl and Beitz model such cases are recognized and handled in the step: "firm up concept variants" (indicated by the arrow in Figure 6). In our approach we view this as a kind of refinement.

### 3.4 Issues related to design objectives

A design objective represents a desired characteristic of a design artifact. The characteristic may have to do with (1) functionality, (2) aesthetics, (3) cost, (4) constructability or manufacturability, (5) disposability, or other concerns. For example, in building design, the objective is to provide an adequate environment for the different activities to be supported by the building [28]. Each activity has certain spatial, lighting, acoustical and other needs. There are interactions between the activities requiring the movement of people and materials among the activities. One of the design objectives is to facilitate this movement (circulation). Typically, the building must meet certain budget requirements. The various objectives have different priorities and it is the task of the designer to ensure that the final design meets the objectives in a satisfactory way. In attempting to formalise the notion of an objective, the following questions need to be answered:

- How to define the characteristic represented by the objective?
- How to express priorities among objectives?
- How to define the concept of satisfaction?

The following sections address these issues.

#### 3.4.1 Objective Definitions

As indicated **earlier**, an objective represents some desired characteristic of the design artifact. The definition of an objective indicates how to compute the amount of the characteristic in the design < determine if the objective is satisfied in a design. The satisfaction of an objective is determined In comparing this value to the desired value (see 3.4.3). We will look at three issues: 1) measurement 2) subjectivity, and 3) computability.

We need to find a measure for the characteristic that the objective represents. For example the adequacy of the environment provided by a building design may be measured in terms of ?h. productivity of the activities supported by the building. The productivity of the environment i-

dependent on a number of factors such as, the sizes of the spaces, the circulation between the spaces, lighting, acoustics, and air quality. These may be considered as sub-objectives of the overall performance objective. Thus, in order to achieve the overall performance objective we need to achieve certain levels of the sub-objectives. The sizes of the spaces may be expressed in terms of the areas or volumes required to contain equipment and people. The circulation between spaces may be expressed either: (1) geometrically as a distance, or 2) topologically in terms of adjacency. The level of lighting depends on the quality and the quantity of light. A measure of the quantity (or more specifically intensity) of light is footcandles<sup>1</sup>. There are two main sources of lighting in buildings: 1) daylight, and 2) artificial lighting. Daylight is usually expressed in terms of the daylight factor<sup>2</sup>. Similarly the acoustical objectives may be defined as providing sufficient levels of insulation against disturbing noises. These levels may be expressed in terms of the intensity (measured in decibels, dB) and the frequency of sound (measured in Hertz, Hz). So far, we have discussed measures that are quantitative in nature. For some characteristics (e.g., view), it may be possible only to indicate more or less of the characteristic, i.e., there is no notion of how much more or less.

In general, there are four scales of measurement: **nominal, ordinal**, interval and ratio [24]. The nominal scale simply classifies objects based on whether they have identical amounts of the characteristic measured (the only operation possible on this scale is **equality**). The ordinal scale has no unit of measurement but has the notion of order, i.e., there is more or less of the characteristic. The interval and ratio scales both have units of measurement, but the difference is that the interval scale has no absolute zero point (cannot determine the equality of ratios). Characteristics such as views can be measured only on an ordinal scale.

Some characteristics are difficult to express objectively, such as those dealing with aesthetic considerations. For example, the objectives requiring good views to the outside from within a space in the building. What constitutes a good view? It is an extremely subjective notion. Due to their subjective-nature such objectives can be evaluated only by the person who specifies the requirement (i.e., the people who will be using the space in the building).

The computability of an objective concerns the following question: Given a final design, is it possible to give a procedure that will determine whether the design meets the objective? For example, consider an objective concerning structural safety with respect to given loads. An analysis of the structure subject to the loads will determine the stresses, which, if below the allowable limits for the material, imply that the objective is satisfied. Sometimes it is difficult to give a computable definition for an objective, usually for characteristics that are subjective in nature.

Sometimes objectives are described by means of an example: identify a kitchen from description of completed kitchen designs (as may be found in catalogues or magazines) [19]. In this case, the objectives are **expressed** in terms of final design features. If we knew the intent behind every decision in the completed design then we could identify the objectives as follows: Consider an **example design containing** the decisions  $D=(d_1, \dots, d_n, \dots)$ . For each decision we can identify a set of objectives **representing** the intent for the decisions  $(O_1, \dots, O_n, \dots)$ . By either accepting or rejecting a **decision**, the user expresses the desire to include some objectives in the present design. This provides yet another motivation for recording design intent.

Another situation where examples may be used is for acoustic specifications. For example.

<sup>1</sup>A footcandle is the amount of light cast by a candle on a square foot area at a distance of a foot from the candle

<sup>2</sup>The daylight factor at a point in a space, is the ratio (in percent) of the illumination at that point to the illumination under the unobstructed sky under certain sky conditions

the designer might require that the noise environment in the building (or a part of it) be like that found along secluded beaches, forests or quiet countrysides. One way to model this would be to measure properties such as the intensity and frequency of sound in these locations and use these values as the design objectives for the building. However, these are only an approximation to the designer's objective because the intensity and frequency of sound may not be sufficient to express the noise environment at secluded beaches and quiet countrysides.

In summary, there may be two problems in defining objectives: 1) subjectivity may lead to the lack of a unique definition, and, 2) it may not be possible to give a computable definition where the user must resort to examples to express an objective(s). In this case, we create a model of the objective(s) based on the examples.

### 3.4.2 Relative Importance

There are different priorities associated with different objectives. Usually, it is not possible to satisfy all the objectives (the notion of satisfaction is defined in section 3.4.3). Hence, trade-offs must be made. Loosely speaking, objectives with lower priorities (less important) are less satisfied than the objectives with higher priorities (more important). There are many techniques used to express this notion in the decision sciences [2]. They include:

- Standard level: A minimum or maximum level of achievement for the characteristic is provided.
- Ordinal: A ranking of the objectives from the most important to least important is established.
- Cardinal: A numeric weight is associated with each objective indicating its importance.
- Marginal: The tradeoffs between objectives are made explicit using either: 1) marginal rate of substitution or 2) indifference curves.

The importance of objectives is often expressed in an ordinal manner because determining the weights or indifference curves is not an easy task. In this thesis we will focus primarily on the ordinal approach to specifying importance. Numerical weights may be used when they can be readily specified.

### 3.4.3 Satisfaction

An objective represents a desired characteristic of the design. One can say that it is satisfied in the current design, if the value of the characteristic in the current design meets certain criteria.

**Criteria for single objectives.** The possible kinds of criteria are the following:

- Inequality: The objective is satisfied in the current design if the value of the characteristic in the current design is greater (less) than some specified value.
- Equality: The objective is satisfied in the current design if the value of the characteristic in the current design is exactly the same as a specified value.
- Maximum(Minimum): The objective is satisfied in the current design if the value of the characteristic in the current design is the maximum (minimum) possible value that can be obtained while satisfying all the other objectives in the current design.

- Increase(Reduce): This is a less rigorous definition of maximum and minimum than the one given above. Because of the complexity of design problems it may not be possible to achieve a true minimum in many cases. For example, consider the problem of minimizing the cost of a building. Whether or not the final solution represents a true (global or local) minimum depends on the strategy used to achieve the objective. So minimizing initial cost might be achieved by minimizing material cost, design cost, and construction cost independently. The assumption of independence is difficult to justify. Thus, we would say reduce initial cost and reserve the use of maximum or minimum for those situations when a truly optimal solution is intended.

**Criteria for Multiple Objectives with priorities.** When priorities are expressed for a group of objectives, the satisfaction criteria are different. They are related to the manner in which the priority information is expressed. In multi-objective optimization, a distinction is made between two kinds of problems: 1) discrete alternatives, and, 2) implicit alternatives, where the alternatives are expressed by constraints over decision variables and may be infinite in number if the solution space is continuous. The implicit alternative case occurs less frequently in a design situation because it requires well quantified objectives and constraints. We recognize the occurrence of this case and it is treated as a refinement (See Section 4.3). We will be concerned more with the case of discrete alternatives.

As seen earlier, design characteristics may be measured on different scales and expressed in different units. In comparing these alternatives it becomes necessary to reduce them to a common scale.

Based on the type of priority information there are a number of ways in which the multi-objective optimum may be defined [2]. We will focus primarily on the lexicographic optimum concept. The lexicographic approach can deal with incommensurate units.

### 3.5 Issues for a computer-based implementation

The use of this methodology in a computer-based environment raises a number of issues:

- Representation
  - The focus of this research is to identify and use computable representations for all entities in the model. However, in practice, it might be difficult to give computable definitions for some objectives. In view of this, we also allow informal textual representation\* The integration of such representations with formal representations needs to be explored
  - **We need** to identify a set of domain specific primitives to express the design at different levels of abstraction. The final design (e.g., a building) is often expressed as engineering **drawings** using lines, curves and other geometric entities and textual annotations to indicate material specifications. In expressing high level objectives it is often desirable to group these geometric and material specifications into meaningful abstractions such as openings (doors, windows), boundaries (walls, roofs) and activity spaces. Objectives and operators will use this representation.
- Operation

- **In this approach we do not expect all the objectives** to be defined at the start of the process. **It should be possible to add and delete objectives** during the course of the design process.
- **The problem of focus, i.e., sequence in which objectives are addressed** needs to be determined. **At the current time we rely on the designer** to solve this problem.
- It should be possible to define design operators as the need arises. We expect to create a library of such operators (**and objectives**) from which appropriate operators can be retrieved and used. A classification scheme **for** the operators is needed to allow a user or the system to browse through **the library** to find the appropriate operator. The granularity of the operators, and hence of **the design** decisions, must be determined. The appropriate level of granularity will depend on the domain.
- Recording
  - A representation for storing the history of this process must be developed. It can be used to support the tasks of explanation and identifying the possible ramifications of changes to the design.
  - The problem of scaling to realistic design problems, with a large number of objective\*, decisions and many intermediate design states, might be a problem.

## 4 Computable Representation of Design Intent

**This section** describes the computable approach taken for representing the components of the design model described earlier. The first section describes how the design and related data are represented. **The second section** describes how design objectives, importance information and assumptions are represented. The last section describes the design process model given these representations.

### 4.1 Representation of Design Data

**All the data** in a design problem are represented as variables. Thus, the building, an activity **within** the building, the material of an opening, the value of bending moment in a structural element the cost of a boundary etc., are all **variables**.

**Variables** may be classified **along** two dimensions:

- **Exogenous and** Design variables
- Entity **and** Characteristic

**Exogenous Variables** are those variables whose values are known beforehand. For example, in **building** design situations climate data, material and labor cost data are examples of **exogenous variables**. They are distinguished from other variables, called design variables, whose values are assigned over the course of the design process.

| Characteristic | Measurement scale | Unit    | Computational type |
|----------------|-------------------|---------|--------------------|
| Cost           | Ratio             | Dollars | Real               |
| View           | Ordinal           | -       | Char string        |

Table 1: Design Characteristics

**Design Entities and Characteristics.** Design variables may be further separated into two categories: 1) Entity - this is a variable type that represents a physical object described by geometry only, or both geometry and material; 2) Characteristics - this is variable type that represents a property measured over these entities (and other exogenous variables). The property may be measured on different scales and units as shown in Table 1.

**Type Relationships.** We use the concept of a type to specify the domain for the different kinds of variables in our design problem. Design characteristics are simple types (reals, strings etc.). Types of entity variables are related in one of two ways: 1) Aggregation relationships, or, 2) Generalization relationships.

In aggregation relationships types form parts of other types. An example of an aggregation hierarchy is shown in Figure 7. The type building that is made up of types: space, boundary, structure, PLEC(Power, Lighting, Electrical and Communication) and HVAC etc.

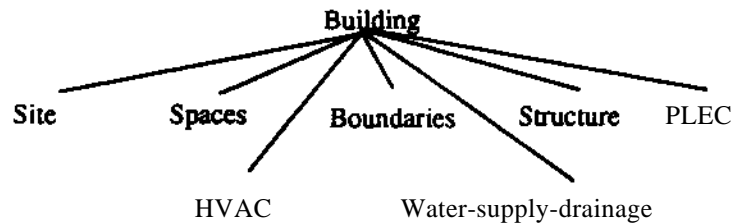


Figure 7: Example of an aggregation hierarchy

Some types are specialisations or generalizations of other types. For example, a door is a specialisation of opening type. An example of a generalization/specialization type hierarchy is shown in Figure 8.

Starting with the most general types (e.g opening) at the root of a specialization hierarchy the more specialized types (e.g., door) can be modelled as the most general type with some constraint added to them. For example, if a type T1 is a specialization of another type T2, this implies that T2 must satisfy some additional constraints as shown in Figure 9. For example, a door is a specialization of an opening, where the specialization constraints include: 1) operable(Opening) (can be opened and closed). 2) height of Opening must be greater than (5 feet, and  $\leq$ ) the width of  $H_i^*$

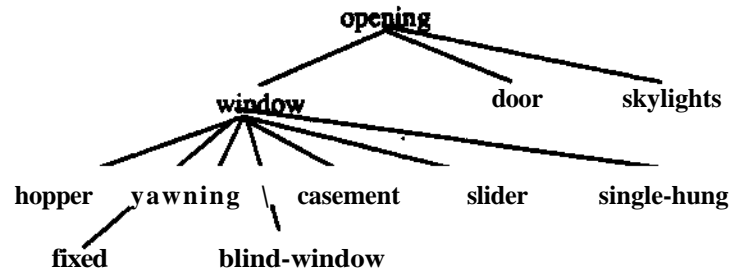


Figure 8: Example of a generalization hierarchy

Opening must be greater than 3 feet<sup>3</sup>. Constraints such as operable(Opening) are very difficult to define. As we will see later the operable() constraint need not be defined. It is satisfied in this design paradigm by a special kind of refinement called refinement by specialization (See Section 4.3).

**Validity Constraints.** Some entity types have a set of constraints associated with them that specify the conditions that the any instance of the type must satisfy to be valid. For the domain of engineering artifacts these constraints are related to the geometry of the artifact. Typically, they specify the valid geometries for the type of object. For example, suppose a rectangular cuboid is specified by the coordinates of its lower left corner (Xll, Yll, Zll) and by the coordinates of its upper right corner (Xur, Yur, Zur). The corresponding validity constraints are:  $X_{ll} < X_{ur}$ ,  $Y_{ll} < Y_{ur}$ , and  $Z_{ll} < Z_{ur}$ . These constraints are added to the design whenever an instance of such a type is introduced in a design situation.

## 4.2 Design Objectives, Importance and Assumptions

An objective is a desired relationship over variables. Objectives may be classified along two dimensions:

- Design goals vs Representation constraints
- Reduce vs non-reduce objectives

**Objectives** may be distinguished on the basis of their source: 1) Design goals, constraints and criteria, **and** 2) Representation Constraints. Each is discussed in the following paragraphs.

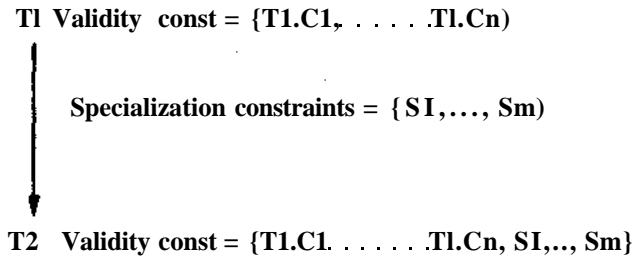
**Design Goals, Constraints** and Criteria include functionality, constructability, cost etc. For a specific domain, such as buildings, these classes may be organized as shown in Table 2. The particular classification shown here is borrowed from Hartkopf et. al. [28]. Some of the performance criteria presented above are difficult to define. But it is possible to define many other criteria. Two example

<sup>3</sup>1 have chosen these values for illustration purposes only

|                    | Physiological  | Psychological   | Sociological  | Economic   |
|--------------------|--|---|---|--|
| Spatial            | Ergonomic comfort<br>Handicap Access<br>Functional Servicing       | Habitability<br>Beauty, calm<br>Excitement, View            | Wayfinding<br>Adjacencies                                   | Space conservation<br>First and<br>Running Costs                                     |
| Thermal            | No numbness<br>No drowsiness<br><br>Heat stroke                    | Healthy plants<br>Sense of warmth<br><br>Individual control | Flexibility to dress<br>w/the Custom                        | Energy conservation<br>First and<br>running costs<br>Cost effectiveness              |
| Air quality        | Air purity<br>No Lung problems<br><br>Cancers                      | Healthy plants<br>Not closed in<br><br>Stuffy feeling       | No irritation<br>from neighbors<br><br><b>Smoke, Smells</b> | Energy conservation<br>First and<br>running costs<br>Cost effectiveness              |
| Acoustical         | No Hearing damage<br><br>Speech clarity<br>Music enjoyment         | Quiet, Soothing<br><br>"Alive"                              | Privacy<br><br>Communication                                | First and<br>running costs<br>cost effectiveness                                     |
| Visual             | No glare<br>Good task illumination<br><br>No fatigue               | Cheerfulness<br>Intimate, Spacious<br><br>Alive             | Status of window<br>First and<br><br>Daylit office          | Energy conservation<br><br>running costs<br>cost effectiveness                       |
| Building Integrity | Fire safety<br>Struct, safety<br>Weathertightness<br>No outgassing | Durability<br>Sense of stability<br>Image                   | Status/Appearance<br>Quality of Const.<br>"Craftsmanship"   | Material/Labor<br>Conservation<br>First, and<br>running costs<br>Cost effectiveil*VH |

Table 2: Organizing Building Performance Criteria: Hartkopf et al.





**Example:** Opening ————— • Door  
**operable(Opening),**  
**Opening.ht > 6 ft**  
**Opening.width > 3 ft**

Figure 9: Specialization types

are provided below.

**Example 1.** This example defines an objective concerning a circulation requirement between two activity spaces in a building. Circulation between two activity spaces implies that there must be an opening of height greater than 7 feet, width greater than 3 feet, and, at the height of the floor on the common boundary between the two spaces. This objective may be instantiated in a design by stating: *circulation(Kitchen, Dining)* where Kitchen and Dining are variables of type activity space. If Kitchen and/or Dining are new variables then the corresponding validity constraints will also be introduced.

**Example 2.** In this example, we will describe an objective that expresses the requirement for a certain amount of daylight in a space. The amount is expressed in terms of the daylight factor which is the ratio of the illumination at the reference point to the outside illumination under the given sky conditions and obstructions [15]. The British standard code of practice recommends daylight factors for different building spaces in the following way: Kitchen - recommended daylight factor not less than 2% over at least 50% of the floor area. The German standard suggest two reference point> **for the measurement** of daylight conditions in rooms in residential buildings as shown in Figure 1- In this **example** a single reference point (typically the middle of a room) is used for the specification of the daylight factor (see Figure 26). The objective constrains the daylight factor at the refereno-point in an activity space to be within the limits LL and UL. This objective may be instantiate! in a design problem by stating: *daylight-factorfAL Ref-point, Sky-condition. Obstructions, LLJ'L>'. It must be emphasized that definition of the objective implies that given the values for the activit> space A1 and the other variables it is possible to determine whether the objective is satisfied, i.r. it*

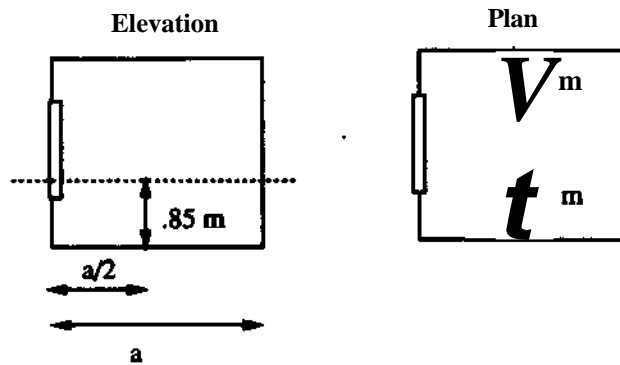


Figure 10: Reference points for daylight (after DIN 5034)

the daylight factor is within LL and UL.

**Library of Objective Classes.** A library of such objective classes might be defined from which the designer may instantiate a set of objectives to define a particular design problem. The designer can define new classes if it is not possible to find an objective class in the library to express a particular requirement.

Representation constraints include the following:

- part-of (aggregation relationships)
- specialization (specialization relationships)
- validity

Validity and specialization constraints have been discussed earlier. The part-of constraints arise from the decomposition of an abstract object or from the composition to a higher level of abstraction. For example, consider the refinement of the circulation objective as shown in Figure 11. The two part-of() constraints are generated due to the aggregation relationships between the activity space and boundary variable types. These constraints are enforced in the final design representation shown in Figure 12. In Figure 12 boundary B7 is shared by both the activity spaces A1 and A2. B7 is the value to which the variable B gets instantiated during the design process.

**Reduce vs non-reduce objectives.** The reduce type of objective involves increasing or derating without limit (I am using all these other words to avoid saying maximum or minimum) of SOME design variable. In this case, there is no "yes or no" answer to the question "is this objective satisfied?"; for example, consider the objective to reduce the cost of a building. In a typical H&A situation, the objective will be achieved via a number of local optimizations (see Figure 13). (Ji)

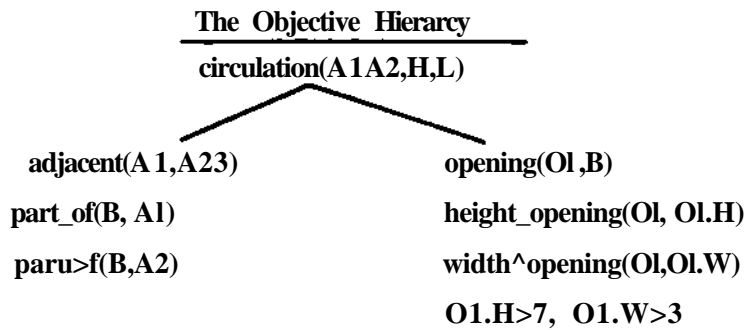


Figure 11: Refinement of Circulation objective

the design solution, i.e., a building the only way of showing that the building satisfies the objective is by describing how the objective was refined, i.e., all of the local minimizations. Also, upon evaluation such an objective type returns the value of the characteristic being increased or reduced under the current variable bindings.

The non-reduce type of objective includes all other objectives which return a "yes or no" answer upon evaluation under the current variable bindings.

**Status of Objectives.** At any time during the design process an objective may have any one of the following status: 1) Active, 2) In-focus, 3) Refined-but-active, 4) Satisfied, 5) Eliminated (or inactive). When an objective is introduced in a design it has the status "Active". The status of the objective changes over the course of the design process, finally resulting in either a satisfied or an eliminated status.

**Importance Groups.** Two or more objectives may be grouped together and importance may be assigned to the members of this group relative to each other in two ways: 1) Ordinal ranking, or 2) Weights.

**Assumptions.** Assumptions are represented in the same way as objectives. The only difference is that there is no **strong requirement that** a particular assumption hold. For example, in sizing openings for **daylight, assumptions** may be made about the reflectances of interior boundaries of a space. Different assumptions may produce different alternatives for the same objectives.

**Status of Assumptions.** As with objectives it is useful to associate a status with assumption\*. The status of an assumptions may be one of: 1) Active, 2) Satisfied, or 3) Eliminated. While an objective participating in a refinement is no longer active after the refinement (its status is either "refined-but-active" or "satisfied"), an assumption may remain active after a refinement. For example, consider the assumption about reflectances of the interior boundaries of a space made to determine the size of an opening for daylight. After the size of the opening is determined the

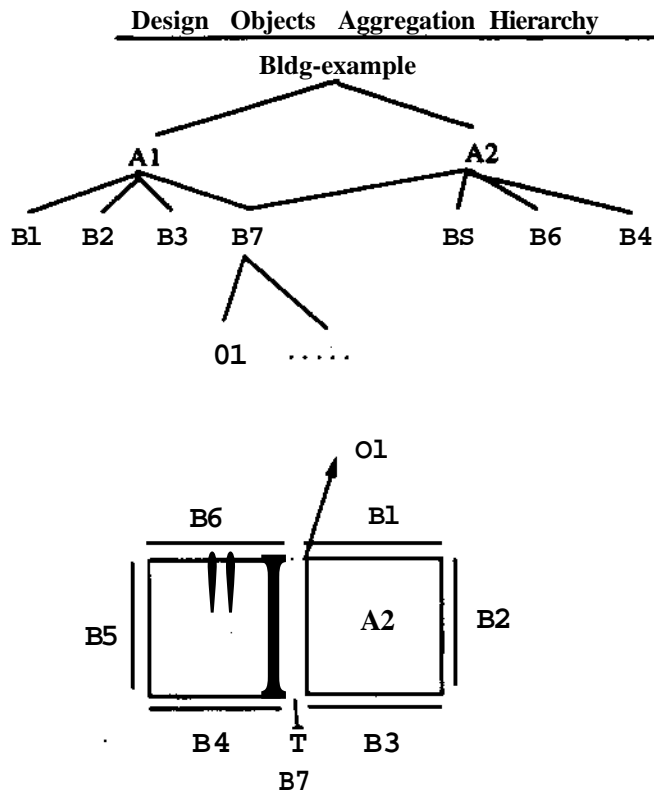


Figure 12: Example of design representation

assumption remains active because the material choices to satisfy the assumption have not been determined. An active assumption is treated just like an active objective.

**Assumptions and subproblem interactions.** Assumptions are the mechanism by which subproblem interactions are handled, i.e., those interactions that are not known beforehand. A subset of the objectives (defining some subproblem) may be refined by making assumptions regarding a subset of variable values that other objectives (defining some other subproblem) may be concerned with. These assumptions become additional objectives that must be met by any solutions to the other subproblem. If the assumptions conflict with the objectives then backtracking and redesign may be necessary.

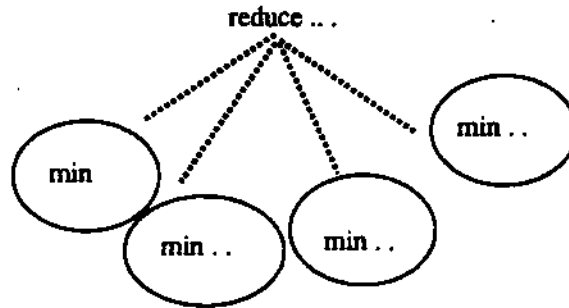


Figure 13: Satisfaction of Reduce type of objectives

### 4.3 The Design Process

The process is modelled in terms of the following activities: 1) Focus, 2) Refine, 3) Evaluate, 4) Select, 5) Resolve. In the discussion of the design process we will make reference to the 'system'. This is to be interpreted as a computer-based environment which records the model of intent built by the designer and provides some assistance to the designer (generating alternatives, evaluating and selecting alternatives etc) as will be seen in the following discussion.

#### 4.3.1 Focus

This is the process of choosing a set of objectives from the currently active set of objectives in the design. The status of the objectives that are picked is changed to "in-focus".

#### 4.3.2 Refine

The process of refinement of an objective(s) is the process of generating alternatives that achieve the objective(s) in focus. The alternatives may 1) establish sub-objectives that must be satisfied to achieve the parent objective(s) or 2) produce bindings for the variables involved such that the objective(s) in focus are satisfied.

**Refinement Generating Subobjectives.** This may be stated as follows:

$$O_1(V_1) \dots O_n(V_n) \wedge A_i(V_{a_i}) \dots \wedge M_r(V_{a_r})$$

$$\text{---}$$

$$O_{xk}, O_{2k}, \dots, O_{mk}$$

where there are  $k = 1 \dots p$  alternatives. Each alternative  $k$  is a conjunction of  $m_k$  objectives. Also, a subset of the variables  $\{y_1, \dots, y_n, x_1, \dots, x_p\}$  that were previously unbound may be assumed to be bound.  $A_i(V_{a_i}), \dots, M_r(V_{a_r})$  are assumptions that are common to all alternatives. Assumptions may also be specific to each alternative. After a process of evaluation and selection out of the

is included in **the design**. The inclusion of this alternative will introduce new objectives and variables into the design.

**Example.** Consider an objective that requires the sound level in an activity space *Al* to be less than some value, *Lad*. The objective may be stated as: *ambtenUsound-level(Al, Acoustic-obs. X-source. Lad)* where *Acoustic-obs* is a variable of type acoustic obstruction list; *N-source* is a variable describing the noise source in terms of its location, geometry and sound level. Sources of noise can be traffic noise, noise from adjacent buildings, or noise from other activity spaces in the same building. Since the human ear responds differently to different frequencies, the sound levels must be considered as a function of frequency. One way to specify ambient noise levels is to use noise-criteria curves [4]. A noise criteria curve is a plot of frequency vs. intensity for a particular range of intensities. An activity space is said to satisfy a particular NC curve if the sound spectrum (plot of frequency vs. intensity) in that space lies below the specified NC curve. However, in this example a single dB value is used to specify the sound levels. The basic strategies for noise reduction are to use either the distance or acoustic obstructions or both. Thus, one strategy that is applicable is to place the activity space at a distance *D* from the source so that the level of intensity of the source becomes the same as the allowable level at the activity space over the distance *D*. Another strategy might be to use acoustic obstructions between the source and the activity space that have a sound transmission loss sufficient to reduce the source sound level to the allowable value in the activity space. They may be expressed as alternative sub-objectives:

- Alt 1: distance(*Al*, *N-source*, *D*),  $D > \sqrt{D_s * D_s * (\text{antilog}((L_s - 160)/10) / \text{antilog}((L_{act} - 160)/10))}$
- Alt 2: blocks-path(*Al*, *N-source*, *Aobs*), lower-limit-sound-trans-loss(*Aobs*,  $L_s - L_{act}$ ).

**Changing Status of Objectives.** All of the new objectives introduced by a refinement have status "active" by default. After the selection process (see Sections 4.3.3 and 4.3.4) has been completed and the selected alternative has been included in the design, the objectives that were in focus (i.e., parents of the refinements) have the status changed to "refined-but-active"<sup>M</sup>. They will be satisfied only when their refinements have been satisfied.

**Refinement Generating Variable Bindings only.** This may be stated as follows.

$$O_1(V_1) \dots O_n(V_n) \text{ } \% \text{ } AM_i(V_{a_x}) \% \dots \% \text{ } M_r(V_{a_r})$$

—  $V'_k$  for  $k = 1 \dots p$

where  $V'_k$  is a set of bindings for all the unbound variables in  $V_1, \dots, V_n, V_{a_1}, \dots, V_{a_r}$ . Assumption may be specific to an alternative. After a process of evaluation and selection one of the alternatives is included in **the design**.

**Example.** Suppose at some stage in the design of a building we have focussed on the following objectives to determine the spatial layout of the spaces: *outside-adjacent(Kitchen, Bldg-env. H1: outside-adjacent(Bed, Bldg-env. B2) outside-adjacent(Living, Bldg-env. B))* .....  
 With some additional assumptions: *left-of(Kitchen, Dining), back-of(Bed, Bath), height(Kitchen 10)* .....

|            |       |    |                               |    |       |
|------------|-------|----|-------------------------------|----|-------|
| <b>Obj</b> | $O_i$ | .. | <b><math>\bullet j</math></b> | .. | $O_n$ |
| $O_n$      |       |    | ..                            |    |       |
| $O_{2k}$   |       |    | ..                            |    |       |
| ...        |       |    | ..                            |    |       |
| $O_{ik}$   | ..    |    | $w_{ij}$                      |    |       |
| ...        |       |    | ..                            |    |       |
| $O_{mk}$   |       |    | ..                            |    |       |

Table 3: Matrix of importance information

it is possible to generate a spatial layout alternative shown in Figure 14. We have determined one set of bindings for the geometry of the spaces and their boundaries based on the objectives and assumptions.

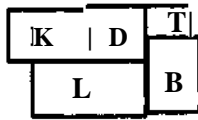


Figure 14: Generating an Alternative

**Changing Status of Objectives.** After the selection process (see Sections 4.3.3 and 4.3.4) has been complete and the selected alternative has been included in the design, the objectives that were in focus (i.e., parents of the refinements) have the status changed to "satisfied".

**Propagation of Importance.** When objectives with associated importance information are refined the designer must indicate how the importance is propagated to the sub-objectives. This depends on how the importance was originally specified.

**Importance specified using weights.** For objectives where importance is expressed using "weight" the proportion of the weight to be propagated to each child must be specified for each of the alternatives. This may be described in the form of a matrix as shown below: In the Table 3.  $O_1, \dots, O_n$  are the objectives being refined;  $O_{2k}, \dots, O_{m^*}$  are the objectives (child objective) in the sub-alternative  $k$ . The designer indicates the weight  $w^k$  for child  $O_{i^*}$  with respect to the parent  $O_i$ . Then the weight of the child may be computed as the sum of all these proportions from each parent in  $k$ . This idea was formalized by Saaty in his principle of hierarchic composition which is stated as follows [22]: Given two finite sets  $S$  and  $T$ , let  $S$  be a set of properties and  $T$  a set of objects with properties as characteristics. A numerical weight,  $w_j > 0, j = 1, \dots, n$ , is associated with each object  $t_j \in T$  such that  $\sum_{j=1}^n w_j = 1$ . Let  $tr_i > 0, i = 1, \dots, m$ , and  $\sum_{i=1}^m w_{ij} = 1$  be the weights associated with /

$i=1, \dots, m \in T$ , relative to  $S_j$ . Then,  $W_i$ , where,

$$W_i = \sum_{j=1}^n w_{ij} w_j$$

gives the weight of  $i$ ; with respect to  $S$ . In our case,  $S$  is the set of parent objectives and  $T$  is the children objectives.

**Importance specified using ordinal ranking.** For objectives where importance is specified using an ordinal ranking, the ordinal preferences for each child with respect to each parent must be specified. Using the same sets  $S$  and  $T$  as before, let  $O_j$  be a ranking associated with each  $S_j \in S$ . Let  $O_{ij}$ ,  $i = 1 \dots m$ , be a ranking associated with  $t_i \in T$ , relative to  $S_j$ . This information is specified by the designer in the same format as for the weights (See Table 3). Then, the ranking  $O_i$  for  $T$  with respect to  $S$  is determined as follows:  $O_a > O_b$  if and only if for  $j = 1 \dots n$ ,  $o_{aj} > o_{bj}$ .

Some kinds of refinement require additional discussion:

- Structural Refinement
- Reduce to min refinement
- Refinement by specialization

**Structural refinement.** In the design process as objectives are generated there is a possibility of redundant objectives. The simplest example would be arithmetic relations like  $X > 5$  and  $X > 6$  which could be reduced to  $X > 6$ . There are more interesting cases such as the example shown in Figure 15. In this case, this refinement implies a kind of "multiple-functionality". Each of the openings O1

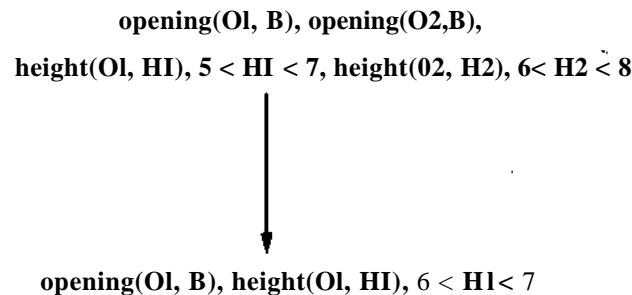


Figure 15: Structural Refinement

and O2 may be required for different purposes, e.g., for daylight or circulation. The reasons for the existence of the objectives can be determined easily by looking for their parents (i.e., the objectives whose refinement generated them). It must be noted that the variables are not yet bound. The terminology structural refinement is used because such refinements do not generate any new objectives.



**Reduce(increase) to Min(max).** This is the refinement by which the semi-formal notion of reduce are converted into formal notion of min. See Figure 16. It must be noted that the trans-

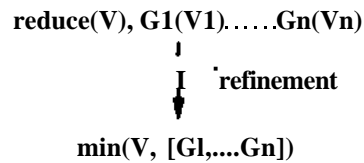


Figure 16: Reduce to Min

formation to a minimization problem is not as simple as the **Figure 16** suggests. In this thesis we will not attempt to deal with this transformation problem. Some aspects of this transformation are of no interest from the point of view of recording design intent (e.g., using integer variables to model discrete variable types such as materials). However, some assumptions made in this transformation might be required to explain the design. We rely on the designer to identify these assumptions. A similar refinement is possible when there is more than one reduce (increase) objective (with associated importance information) which could result in some form of a multi-objective optimization problem.

**Specialization Refinement.** The entire design process can be carried out with only the most general types of objects, since the more specialized types can be modelled as the most general type with some constraints added to it. However, the designer may find it convenient to work with specialized types (e.g., door vs opening). Specialization refinement is a facility that allows the designer to introduce the more specialized type in the design. The specialized type may be introduced when the appropriate specialization constraints are present in the design as shown in Figure 17. The con-

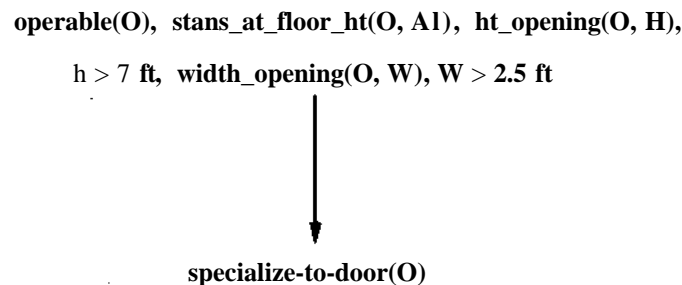


Figure 17: Specialize to door

straints involved in this refinement become validity constraints associated with the more specialize!

type.

In summary, the following points are noted:

- **Importance is not part of the refinement (not hardcoded into the operator).** The justification is that importance seems to be specific to a design situation or a designer. Different designers might associate different importance with the same alternatives and thus it does not make sense to make it a part of the refinement.
- **All the objectives in an importance group must be refined simultaneously.** This makes sense because whenever relative importance is specified between objectives as it implies that there is a potential conflict.
- **When importance is associated with objectives being refined, the importance is propagated from parent objective(s) to the children for each of the alternatives.** The designer must indicate how this occurs by providing information in the form of the importance matrix shown in Table 3.

#### 4.3.3 Evaluate

The evaluate step is the process of evaluating the set of alternatives generated upon refinement with respect to a set of objectives in focus. This set of objectives in focus is different from the ones considered for refinement.

**Focussing for Evaluation.** The first step is to determine the objectives that are to be used in the evaluation of the alternatives. Let  $V_{ait}$  be the variables in the alternatives. Thus  $V_{ait}$  may be the variables whose bindings have been established by refinement or they may be the variables that occur in the subobjectives generated by the refinement. Any active objective which has variables that occur in  $V_{ait}$  becomes a candidate for evaluation.

Depending on the amount of information available during the evaluation we will distinguish between two kinds of evaluation:

- Complete evaluation
- Heuristic evaluation

Complete evaluation occurs when given the current values for the variables it is possible to determine the result of the evaluation with absolute certainty. Heuristic evaluation is carried out only with partial information on variable values and thus it involves a degree of uncertainty. We will not attempt to quantify this uncertainty.

The results of the evaluation of a set of alternatives  $A_1, \dots, A_n$  with respect to  $O_1, \dots, O_m$ , may be expressed in the form of the matrix shown in Table 4 where an entry  $x_{ij}$  indicates the result of the evaluation of alternative  $i$  with respect to objective  $j$ . Depending on the type of objective  $j$  (reduce vs non-reduce) involved the result of the evaluation is different. For an objective of the non-reduce:

| Alt or Obj | $O_x$ | $O_y$ | .. | .. | $O_n$ |
|------------|-------|-------|----|----|-------|
| $A_1$      | HNE   | ANE   | .. | U  | 20    |
| $A_t$      | HE    | ANE   | .. | u  | 10    |
| ...        | HE    | ANE   | .. | u  | 15    |
| ...        | HE    | AE    | .. | u  | 14    |
| $A_k$      | HE    | ANE   | .. | u  | 11    |

Table 4: Matrix showing results of evaluation

$x_{ij}$  = ANE (absolutely not eliminated)  
I AE (absolutely eliminated)  
I HNE (heuristically not eliminated)  
I HE (heuristically eliminated)  
I U (unknown)

where ANE and AE result from a complete evaluation while HNE and HE result from an incomplete evaluation. In the latter case the objective status remains "active". Such an objective must be re-evaluated later when more information is available. The "unknown" case arises when no evaluation (not even a heuristic evaluation) is possible at this time for that objective. For an objective of type reduce:

$x_{ij} = A_{ij}$   
I U (unknown)

An objective of type reduce involves increasing or reducing a variable involved without limit. During the evaluation process, the value of this variable  $A_{ij}$ , representing some characteristic (e.g., material cost of some component), is computed for the alternative  $i$  and objective  $j$ . Again, the "unknown" case arises when no evaluation is possible at this time with respect to that objective. The unknown situation may also arise because the designer might identify objectives that are not defined (<> the system). However, the designer is able to evaluate them with respect to alternatives, i.e., the designer is able to indicate some sort of preference based on this objective.

#### 4.3.4 Select

In this process the information provided by the evaluation matrix is used (along with other information obtained from the designer) to select one alternative from the different alternatives to include in the design. We will distinguish two cases based on whether any importance information is specified or not:

- When importance information is specified the selection process will result, in one of the following:
  - A single "best" alternative is identified
  - A set of equivalent "best" alternatives is identified.

- When no importance information is available the selection process will result in one of the following:
  - A single "best" alternative is identified.
  - A set of non-inferior alternatives is identified.

We consider each of the cases separately.

**Selection with Importance specified.** Consider the following example. It is assumed that the importance is stated by an ordinal ranking and the criteria for evaluation is lexicographic. The evaluation matrix is shown below:

| Alt or Obj | 01  | 02 | 03 | 04   |
|------------|-----|----|----|------|
| A1         | ANE | 10 | 20 | high |
| A2         | ANE | 20 | 30 | med  |
| A3         | ANE | 30 | 25 | low  |
| A4         | AE  | 50 | 10 | high |

Suppose the ordinal ranking is given as follows:  $04 > 03 > 02 > 01$ , i.e., 04 is more important than 03; 03 is more important than 02; 02 is more important than 01. The lexicographically best choice is A1 (A4 is eliminated). The designer might respond by:

- accepting the system choice; or
- rejecting the system choice and making a different choice.

When the designer rejects the system choice, it implies that the model of intent known to the system (i.e., which the designer has built through his or her earlier actions) is inconsistent with the current model in the designer's mind. In other words, the designer has changed his intent. This change may occur in the following ways:

- A change in the relative importance of the objectives, e.g., if 01 was more important than 02 before now the inverse is true.
- The designer is using some unstated objective to guide the selection of alternatives (by which we mean either an objective not introduced to the system or perhaps known to the system but which has not been brought into focus for this evaluation).

These alternatives are presented to the designer who must provide the system with the appropriate information **that** will justify the designer's choice instead of the original system choice.

Going back to our example, suppose the designer rejects the system choice of A1 and picks A2. This choice is justified by:

- Changing importance: 03 becomes the most important objective. In this case, this is the only permutation of the ranking that will justify the choice. In general there may be more than one way to justify the choice.

- Introducing a new objective: Some other unstated objective is the most important and is guiding the choice. The designer must introduce the objective with the related importance information.

Once the choice has been justified both the system and the designer are consistent with respect to their current models of intent.

In the example, the designer may pick the eliminated alternative A4 instead of A2 or A1. This may be justified by making O2 the most important objective (as shown in the table A4 has the highest value for O2). In this case the objective that is causing the elimination of the alternative must be changed or eliminated so that the designer can stay with that choice. This is done in a resolve activity. See Section 4.3.5.

**Selection with No Importance specified.** Consider the same example. Again, it is assumed that the importance is to be established by an ordinal ranking and the criteria for selection is lexicographic. The evaluation matrix is shown below:

| Alt or Obj | O1  | O2 | O3 | O4   |
|------------|-----|----|----|------|
| A1         | ANE | 10 | 20 | high |
| A2         | ANE | 20 | 30 | med  |
| A3         | ANE | 30 | 25 | low  |
| A4         | AE  | 50 | 10 | high |

If no importance information is provided then the system can identify only non-inferior alternatives. In this case, A1, A2 and A3 are all non-inferior and A4 is eliminated. Suppose the designer responds by selecting A1. Given the assumptions about ordinal ranking and lexicographic criteria the designer's choice may be justified by:

- O4 is the most important objective.
- Designer is using some other unstated objective to guide the selection.

In general there could be an infinite number of ways to justify the choice and designer might have intended only one of them. Generating the possible justifications for the designer's choice is the inverse selection problem. It is discussed below.

**The Inverse Selection Problem.** The selection problem may be defined as the problem of identifying an alternative given the evaluation matrix, the selection criteria (e.g., lexicographic minimum) and the importance (e.g., ordinal ranking). The inverse of this problem is that of determining ranking given the choice of alternative, the evaluation matrix and the selection criteria. It may be possible to outline an algorithm that solves this problem at least for the case of ordinal ranking and lexicographic criterion. In other cases it may be not possible (e.g., when weights are involved). However, for our purposes it may be sufficient to recognize an inconsistency between the system model of intent and the designer's model in their mind. We may rely on the designer to corral the inconsistency, i.e., supply the correct importance information that justifies the designer's choice when it conflicts with the system choice.

| Alt or Obj | $O_i$ | $O_j$ | .. | .. | $\bullet P$ |
|------------|-------|-------|----|----|-------------|
| $A_i$      | ANE   | ANE   | .. | .. | AE          |
| $A_j$      | AE    | ANE   | .. | .. | AE          |
| ...        | AE    | ANE   | .. | .. | ANE         |
| ...        | ANE   | AE    | .. | .. | ANE         |
| $A^*$      | ANE   | ANE   | .. | .. | AE          |

Table 5: Matrix showing results of evaluation

**Effect of changing importance on the objective hierarchy.** When the importance associated with some objective has to be changed to correct an inconsistency, the importance associated with the parents (if any) must be reassessed by the designer.

An arbitrary selection occurs when the designer makes a selection without any justification (i.e., no objective used for evaluation and hence no evaluation is performed). This is allowed and recorded as such. After an alternative has been selected it is included in the current design. Depending on the type of alternative (sub-objectives or bindings) the objectives that were in focus for refinement have the status changed to "refined-but-active" or "satisfied". The sub-objectives have status "active".

In summary, the following points are noted:

- Given the evaluation matrix and other information, the system comes up with choices. The designer may or may not agree with them. If the designer disagrees then it implies that the designer has changed his or her original model of intent that is recorded in the system.
- In some special cases, it may be possible for the system to suggest changes that justify the choice. These suggestions are presented to the designer who may verify it. If this is not possible, then the designer must indicate how the model of intent has been changed.

#### 4.3.5 Resolve

A resolve situation occurs due to a potential conflict between some objectives in the design. There are three ways to get into a such a situation:

- no alternatives are generated for refinement;
- all alternatives generated for refinement have been absolutely eliminated by evaluation (see Table 5); or
- during the selection process the designer intentionally selects an eliminated alternative even when other non-eliminated alternatives are available (see Section 4.3.4).

Due to the nature of the design methodology where we allow for arbitrary choices, assumptions and importance there are many ways by which such a situation may be handled:

1. backtrack (or changing a previous selection or changing importance);
2. change or replace assumptions;
3. change or replace objectives;
4. eliminate objectives; or
5. some combination of 2, 3, and 4.

Before describing the different options, some definitions are needed.

**Levels in the Objective Hierarchy.** It is useful to distinguish two levels in the hierarchies shown in Figure 18: 1) Top-level and 2) Leaf level. In Figure 18 objectives 01, 06 and 07 are top-level

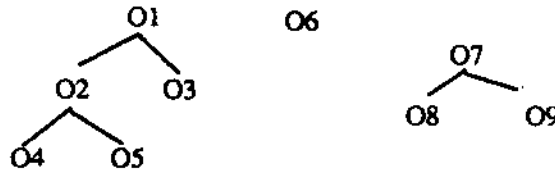


Figure 18: Levels in the objective hierarchy

while 04, 05, 03, 06, 08 and 09 are leaf level objectives. Notice that 06 may be considered to be in both the top and leaf levels.

**Changing an Objective or Assumption.** Changing an objective implies changing the values of the exogenous variables. So for example, in the case of the sound level objective: *ambient-sound-level*(Al, Acoustic-obs, N-source, Lad), N-source and Lact are exogenous variables. N-source is a description of the noise source and Lact is the desired upper limit on the sound level in Al. During the instantiation of the objective in the design Lact may have been given a value, say 60 dB. In other words, the designer desires the sound level in Al to be within 60 dB. During the course of the design this value may be increased or lowered. In this case we say that the objective has been changed.

**Replacing an Objective or Assumption.** Changing exogenous variable values is not sufficient to model all possible changes to an objective or assumption. For example, consider a chain: from reduce-cost() to cost < X. Such changes are modelled by "replace" operation as opposed to "change" operation. In the case of both change and replace a new version of the old objective is created. See Section 5.3 for how these versions are maintained.

**Backtrack.** Backtrack to the previous point, where an arbitrary choice was made (shown in A in Figure 19) and try one of the other alternatives. The activities upto the current state may U

replayed on the other alternative if the operators for refinement, evaluation and selection are known. If significant designer input was involved then we will not be able to proceed without designer assistance.

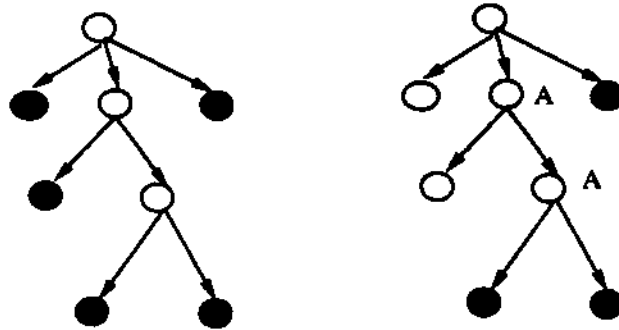


Figure 19: Progression of states leading to resolve

**Change assumptions.** An assumption may have been used in another refinement. See Section 4.2. In changing the assumption it is possible that the other refinement would have to be redone. In other words, all the alternatives generated by the refinement become invalid.

**Change objectives.** Only top-level objectives may be changed. So if an objective at some other lower level has to be changed the designer must traverse the hierarchy until the top-level objective is reached. The effect of changing a top-level objective on the lower level objectives depends on how the objective was refined. Consider the example of the sound level objective: *ambient-sound-level(AI, Acoustic-obs, N-source, Lact)*. The alternatives generated by refinement:

- Alt 1:  $\text{distance}(AI, N\text{-source}, D), D > \sqrt{Ds \cdot Ds} \cdot (\text{antilog}((Ls - 160)/10)/\text{antilog}((Lact - 160)/10))$
- Alt 2:  $\text{blocks-path}(AI, N\text{-source}, A\text{obs}), \text{lower-limit-sound-trans-loss}(A\text{obs}, Ls - Lact)$ .

If Lact is changed from say 60 to 70 dB, then for Alt 1 the limit on D changes, while for Alt 2 the second objective *lower-limit-sound-trans-loss(Aobs, Ls - Lact)* changes. Notice that the first objective for Alt 2 remains unaffected. Any refinements that included only the first objective, but not the second, will remain valid.

**Eliminating an objective.** Only top-level objectives may be eliminated. If the objective is "active" then eliminating it has no effects. The effects of eliminating a satisfied objective on the type of activity that satisfied the objective: 1) refinement or 2) evaluation. Each of these cases are considered below. \*

In case of a refinement activity the following situations may occur:



- The alternatives become invalid, i.e., a different set of alternatives may be generated without the eliminated objective or maybe that no alternatives are possible.
- All the objectives used in refinement, evaluation and selection are "active" again.
- The alternative that was chosen to be included in the design is revoked. This has different implications depending on whether:
  - The alternative is a set of sub-objectives: the corresponding objectives must be eliminated.
  - The alternative is a set of variable bindings - the corresponding variables become "unbound" - this may cause a number of objectives to change status and any refinements that were generated from those bindings become invalid.

In case of an evaluation activity, the status of an alternative may change from eliminated to not-eliminated because the elimination of an objective may free up an alternative that was eliminated due to that objective. An alternative that was selected might become unselected, while an alternative that was previously unselected may become selected. These changes have different implications depending on whether:

- The alternative is a set of objectives - the corresponding objectives must be eliminated: and the objectives corresponding to the newly selected alternative must be included in the design.
- The alternative is a set of variable bindings - the corresponding variables become bound to the new bindings corresponding to the newly selected alternative, which may cause a number of objectives to change status.

**Changing** material and geometry of design entities as a tool for Resolve. In this paradigm the designer is not allowed to directly manipulate the design. The designer is forced to work with the idea of refinement of objectives. However, it may be useful to allow a designer to alter the variables at the lowest level of abstraction in the type hierarchy (i.e., variables representing material and geometry of design entities) in a "what-if scenario to identify the objectives that are violated. This will help the designer determine what should be done to the objectives to obtain the desired result. This assumes that the designer has generated at least one design solution and the solution does not satisfy the designer.

**Variable Change Propagation.** Even though a design variable cannot be directly altered by the designer, it can be changed during a "what-if scenario or as a consequence of deleting or changing objectives. Consider the hierarchies shown in Figure 20. The highest level objectives, i.e., at level 0 are established by the designer. These objectives will often involve the most abstract variable,  $D1^0$  in Figure 20. For example,  $D1^0$  could be the building, and  $D1^1$  to  $Dp^1$  could be the variables representing activity-spacer, structural system, HVAC system and PLEC system. There may be more than one such abstract variable, although in this example there is only one. The variables  $D1^1$  to  $Dp^1$  are related to the higher level variable  $D1^0$  by part-of() constraints. As the objective hierarchy is refined, the design hierarchy also gets refined. It is not necessary that both hierarchies have the same number of levels as shown in Figure 20. The lowest levels of the design hierarchy, i.e.,  $D^m$  to  $Dp^m$  will be made up of variables representing geometry and material

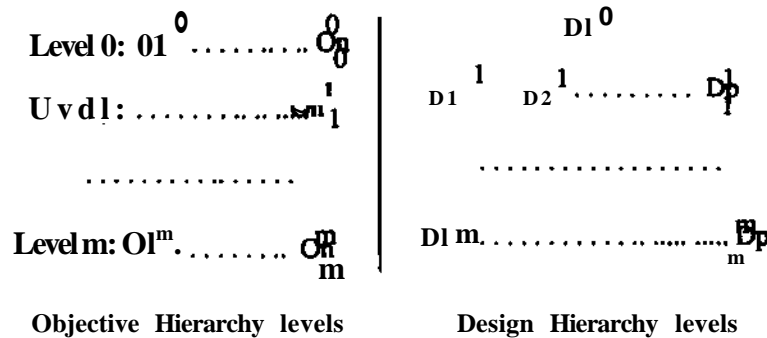


Figure 20: Levels of objective and design hierarchies

an artifact such as a building). Correspondingly, the lowest level objectives will represent geometric relationships (e.g., adjacencies between objects) and material property (e.g., sound transmission loss values) constraints or other characteristics such as cost.

Suppose a subset of the variables  $D_i^m$ ,  $1 < i < p_m$  are changed. A subset of the objectives  $O_i^m$ ,  $1 < i < n$  will be immediately affected. Since these objectives are refinements of corresponding higher level objectives at level  $m-1$  they will also be affected and so on till level 0. Another way of looking at this is from the perspective of the design hierarchy: The variable  $D_i^m$  are part of higher level variables at level  $m-1$ . A change in  $D_i^m$  implies a change in the higher level variables of which  $D_i^m$  are a part.

As a result of one or more of these changes the effect of resolve may be to change the objective by creating new version of an objective - entire objective hierarchies may be altered.

## 5 Recording the history of the Design Process

The history of the design process described in the earlier sections constitutes the record of design intent. The basic unit of the record is the design state. The record is a time sequence of design states. Both of these concepts are described in the following sections. We have adopted a single task **approach**. In other words, no two design activities may occur in parallel. It is conceivable that two refinements occur in parallel if their variable domains are completely or almost distinct. This raises a number of issues because there may be more than one set of variable binding at the same time in the design. However, we will not be concerned with this problem in this ~~thM~~.

### 5.1 The Design State

The following entities occur within the design process model described earlier:

- objectVPS and importance groups;

- **assumptions;**
- design variables;
- exogenous variables;
- alternatives;
- evaluation matrix; and
- design operators (incl. heuristics).

Both the alternatives and evaluation matrix may be redundant information in the sense that they can be generated from the objectives-in-focus, assumptions, variable bindings and the operators. However, some evaluations may be carried out by the designer in which case operators may not be available. During the course of the design process all of these entities can change. Some are changed by the design activities while others maybe changed explicitly by the designer (eg objectives and exogenous variables). The design state  $S_i$  may be represented as follows:

- Objectives:  $O_a$  - active;  $O_s$  - satisfied;  $O_e$  - eliminated;  $O_{ra}$  - refined but. active.
- Assumptions:  $AS_a$  - active;  $AS_s$  - satisfied.
- Design variables:  $Vd$  - current bindings for the variables (some of these variables may be unbound).
- Exogenous variables:  $V_e$  - current bindings, for the exogenous variables (all of these variables are bound).
- Alternatives:  $A_{ae}$  - absolutely eliminated alternatives,  $A_{nt}$  - not-eliminated,  $A_g$  - selected alternative. There may be no alternatives.
- Operators

A particular version of objectives, variables, assumptions and operators are associated with each state. One state is always identified as the current state in the design. This state need not be the latest state in the time sequence of design states.

## 5.2 The Activity Record

A **transformation** from this state  $S_i$  to the next  $S_{i+1}$  occurs as a result of any one of the following actions (see **Figure 21**):

- Focus/Refine/Evaluate/Select
- Resolve
- Modification of objectives by designer

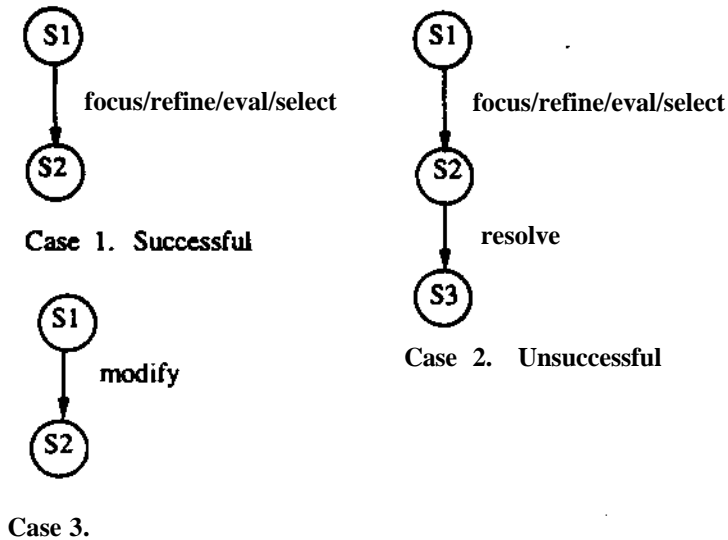


Figure 21: Transformation between Design States

For each of these three cases the activity record is used to maintain information about the activity. The kinds of information generated and recorded for each type of activity are discussed below.

**Focus/Refine/Evaluate/Select.** This activity is carried out as follows:

- Identify objectives in focus for refinement.
- Refine the objectives to generate alternatives.
- Identify objectives in focus for evaluation of the alternatives.
- Perform the evaluation of the alternatives to obtain an evaluation matrix (See Section 4.3.2).
- If all the alternatives are eliminated then the resolve activity (Case 2 in Figure 21) is invoked; else, select an alternative (See Section 4.3.4).
- If an alternative is successfully selected then include it in the design (Case 1 in Figure 21).

Information generated during this kind of activity that must be recorded include: 1) importance matrix, 2) objectives in focus for refinement, 3) operator used for refinement, 4) assumptions made in the refinement, 5) objectives in focus for evaluation, 6) evaluation matrix, 7) operators used in the evaluation, 8) selection criteria, and 9) selected alternative. The importance matrix contains information used in propagating importance in the objective hierarchy (See Propagation of importance in Section 4.3.2).

**Resolve.** As indicated earlier, this case arises when all of the alternatives generated in the focus/refine/eval/select activity are eliminated (Case 2 in Figure 21). The nature of the changes that were made to rectify the potential conflict among objectives are recorded, i.e., backtracking, changing or replacing assumptions, changing, replacing or eliminating objectives, or some combination of the above.

**Modification.** Objectives and operators are the only items that can be explicitly modified by the designer. The modifications considered in this activity are not motivated by conflict (resolve) but by some external influences (e.g., the design problem description may be changed). The particular modifications carried out are recorded, i.e., which objective(s) or operator(s) are modified. The actual forms of the objective(s) and operator(s) are stored as versions with the objective(s) and operator(s) themselves as explained in the following section.

### 5.3 The Organization of the Design Record

The organization of the information in the design record is dependent on how the information is to be used. These include 1) browsing through the record, and 2) determining the effect of modification.

#### **Browsing.**

- Get all instances of a class of objective: eg. adjacency(). or more specifically, adjacency() involving variable X.
- Get all instances of a type of variable: eg. activity space or opening.
- Get all instances of objectives involving a given set of instances of variables.
- Identify person(s) responsible for an objective(s).
- Show all versions for an objective.
- Show all versions of a variable.
- Determine the alternative for a particular version of a variable.
- Determine the most recent state where an arbitrary selection was made.
- Determine the state in which an objective(s) was in focus. This is equivalent to asking for r In-state in **which** the objective status was changed.
- Get **assumptions** made in generating an alternative.
- Show **all** alternatives generated in a state.

Determining the effect of change.

- Changing the value of a leaf level variable and its effects on the objective and variable hierarchies.

- Adding, eliminating, and changing top level objectives.
- Adding, eliminating, and changing assumptions.

The representation of the information can be tailored to efficiently support a given set of requirements. In the following sections, we describe a representation scheme to support the requirements outlined earlier.

### 5.3.1 Objectives

The following pieces of information is represented with each objective: 1) name, 2) content, 3) importance, 4) author, 5) version history, and 6) parents and children.

**Content.** The content of an objective is described by the following items:

- class of objective: identifies the method for evaluation and refinement; and
- variables: instances of the variables that are constrained by the objective

**Importance.** The objective may have importance information specified relative to other objective\*. All these objectives are grouped into a structure called the importance group.

**Author.** This refers to the name of the person introducing the objective (or the particular version of it). In building design it may be a contractor, client, architect etc. This is used only for the highest-level objectives.

**Parents and Children.** The parents are those objectives of which this current objective is a refinement. If it is not the refinement of any objective and has been introduced by the designer then this value is set to *designer*. The children are those objectives that are refinements of this objective

Versions of objectives are created by the "change" and "replace" operations on the objectives. It should be possible to retrieve any version of an objective from a current version. We will follow an approach similar to the one used in the ORION model [10], [1], where a generic version object is  $u \times :1$  to record the version derivation hierarchy (Figure 22). Every version of the objective maintain\* t reference to this object and hence it is possible to retrieve any other version of the objective with-rolling the history back to the state in which the particular version existed.

**The Current Version as a function of state.** A particular version of the objective is associated with each state in the design. For example, 01.VI could be existing from states 0 to 10. 01.V'2 f<>m states 11 to 20 etc. Thus, given a state, say 12, the form of the objective for that state is determin\* -I

**Objective status.** The status of the current version of the objective is also a function of the sm?> (in the same way as above) and may be one of: active, refined-but-active, satisfied or eliminated

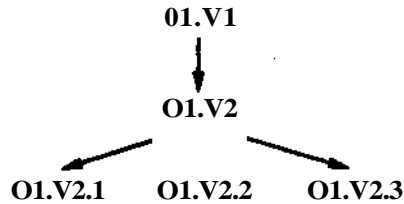


Figure 22: Objective Version derivation hierarchy

### 5.3.2 Assumptions

They are similar to objectives with the following exceptions: 1) the status of an assumption can be one of active or satisfied, 2) assumptions do not have parents or children, and 3) they have no importance associated with them.

### 5.3.3 Variables: Design and Exogenous

The following pieces of information are represented with each variable: 1) name, 2) type, 3) value, and 4) versions.

**Type and Value.** A design variable can be made up of other variables (a composite type):  $V_1 = (V_{11}, V_{12}, \dots, V_{1n})$  where  $V_{1j}$  is related to  $V_1$  by part-of( $V_{1j}, V_1$ ). Alternatively, a variable may be bound to a character string, integer etc (simple types).

**Versions.** Versions are alternative values for the same design variable. In this paradigm alternatives may be generated in the same state by a refinement activity or alternatives may be generated in different states as shown in Figure 23. This happens because the designer chooses to arbitrarily select one alternative in state  $S_1$  and pursues the design till state  $S_6$  generating one alternative for the layout and location of openings. At this point the designer could backtrack to  $S_1$  and pursue a different alternative thereby generating the alternative in state  $S_{10}$  as shown in Figure 23. As with the case with objective versions, it is useful to be able to retrieve any version from a current version of the variable. The same approach used for objectives is adopted for this purpose. A generic version object is used to record the version derivation hierarchy. Every version of the variable maintains a reference to this object and hence it is possible to retrieve any other version of the variable.

**Current Version as a function of state.** The same idea of relating the current version of the variable to the state is also used with respect to variables. For example,  $V_1.v_1$  could be existing from states  $n$  to  $10$ ,  $V_1.v_2$  from states  $11$  to  $20$  etc. Thus, given a state, say  $12$ , the value of the variable in that state is determined. If the variable is unbound in a particular state it is assigned a special symbol indicative of this condition. If a variable becomes non-existent in some state (this could happen due to backtracking) then another special symbol is used to indicate this condition.

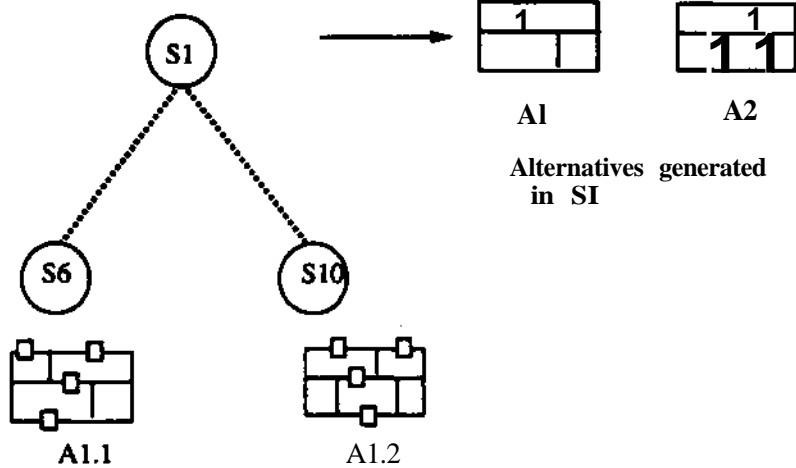


Figure 23: Alternatives generated in different states

Two other items that may be useful to associate with variables are: 1) a change notification flag, and 2) the list of part-of constraints involving the variable (only variables that represent entities can form such constraints). The change notification flag is used for indicating when a variable value is changed (see Section 4.3.5) and hence everything related (determined using the part-of constraints) must be notified.

### 5.3.4 Alternatives

The following pieces of information are represented with each alternative: 1) name, 2) variable bindings and/or objectives, 3) alternative-group, and 4) status.

An alternative may include a set of variable bindings and/or objectives. An alternative made up of variable bindings is represented by maintaining references to the different versions of the variables **involved as shown in Figure 24**. In the case of alternatives that are objectives, the objectives will not be **versions of each other**. They could be completely different objectives as shown in Figure 24. The **alternative maintains** a reference to all of the objectives.

**Alternative group.** It refers to the group to which the set of alternatives belong. All the alternatives in the group were generated in the same state, i.e., the same refinement step. The alternative group contains:

- a reference to the alternatives that form a part of it; and



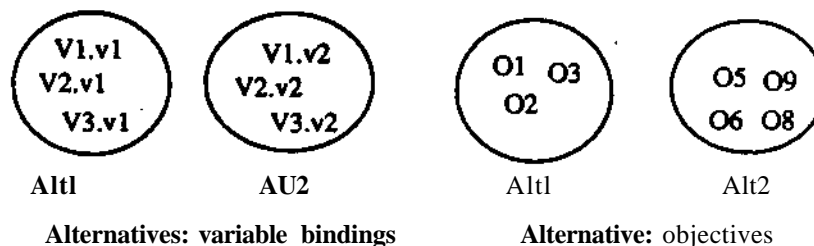


Figure 24: Representation of alternatives

- a reference to the activity record for the activity generating alternatives (or this could be a reference to the particular state in which the alternatives existed).

The activity record is a reference to the focus/refine/eval/select activity in which the alternative was generated.

**Status.** The status maybe one of: eliminated or not-eliminated or selected. The status of the alternative is represented in the same way as the status of objectives, i.e., it is a function of the state.

### 5.3.5 Design Operators

The following pieces of information are represented with each operator: 1) name, 2) content, and 3) versions.

**Content.** Operators are involved in refining or evaluating instances of classes of objectives. They may be associated with a single class or with multiple classes of objectives. From an implementation perspective they could be rules in CLP(R) or a C/Fortran function.

**Versions.** A version is created by changing an operator definition. A change in the operator definition could be motivated by:

- a change in the semantics associated with the objective; or
- a change **in** technology e.g. a new material is invented with better properties, or a  $n^*$ -|| alternative for addressing some objective is discovered.

As was the case with objectives and variables, it is desirable to maintain the entire version history with every version of the operator. The same approach used for objectives and variables is used. A generic version object is used to record the version derivation hierarchy. Every version of the variable maintains a reference to this object. As with objectives and variables an operator version is associated with every state. Thus given the current state, it is possible to determine the current operator version. The change and versioning of operators is not supported in the prototype.

### 5.3.6 The Design Record

The record is a time sequence of states. Because of the possibility of backtracking, two states may be derived from the same state. So the logical structure is tree-like. See Figure 25. The record is

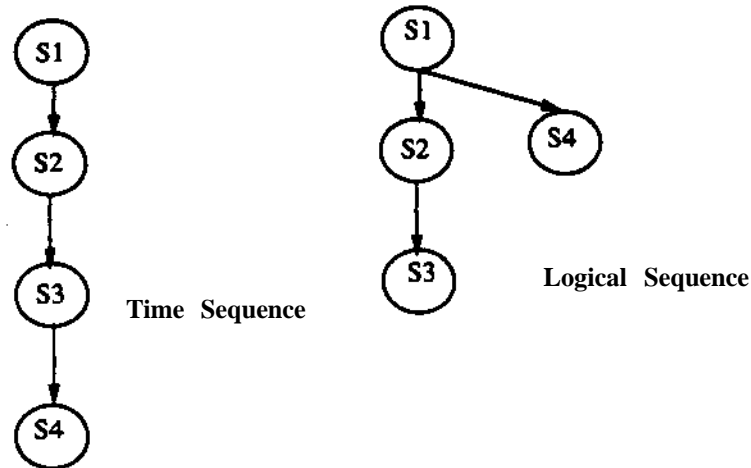


Figure 25: Record: Time vs Logical sequence

implemented as a linked list of states. Each state contains the following information: 1) name or some reference to the state, 2) next-state(s), 3) previous state, 4) activity record involved in the transformation from the previous state to the current state, and 5) a flag indicating whether or not an arbitrary choice for alternative was made in that state.

## 6 Example

This example is concerned with the design of a residential building. The highest level objective\* in building design are to support the activities occurring within the spaces of the building. However for this example, we will assume that the highest level objectives have been refined to objectives\* concerning the spatial, acoustic, visual, thermal and air quality requirements associated with each space. In this example, we will deal only with a subset of these objectives as shown in Table <<

In the Table 6 "a ~" indicates equality with a tolerance. For the area objective, we may specify allowance to be 15 percent (both positive and negative). For the daylight objective, P and mid iulir.u. the position of the reference point within the space at which daylight factor is to be measured. There are many ways to specify this position as shown in Figure.26. Another objective that is considered is to reduce the conducted heat loss from the building, expressed as: reduce-conducted-heat-loss( Bf-k Bldg-env). Other objectives will be included at different stages of the process.

| Activity-Space | Area (sq. ft.) | SL (dB) | Daylight (DF) | Circulation |
|----------------|----------------|---------|---------------|-------------|
| Kitchen(K)     | - 120          | -       | 2,3 P         | -           |
| Dining(D)      | - 100          | -       | 2,3 P         | Kitchen     |
| Living(L)      | ~ 144          | -       | 2,3 P         | Dining, Bed |
| Bed(B)         | - 130          | < 30    | 2,3 P         | -           |
| Bath(T)        | - 50           | -       | 3,4 mid       | Bed         |

Table 6: A subset of objectives for a residential building

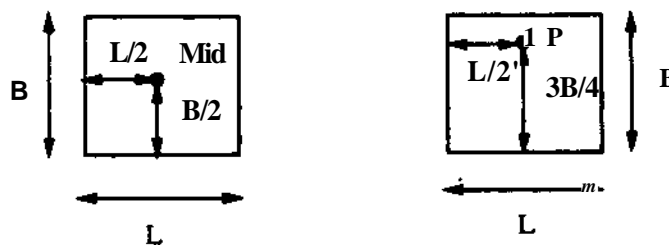


Figure 26: Specifying reference positions

**Environment.** The description of the environment includes the site for the building and the location of the noise source. The intensity of the noise is 80 dB at a distance of 10 feet away. The 97.5 percentile temperature for winter months is 15 F. Other data will be included as needed.

**The Time dimension.** The specifications for the building should include the notion of time—e.g., the time period during a day when the ambient sound levels are desired. However, to simplify the discussion of the example, this dimension has been ignored.

**Representation of Geometry.** The activity spaces, boundaries and openings are assumed to be rectangular cuboids. It is possible to define them in at least two ways: 1) it may be defined by specifying the coordinates of the lower left corner and upper right corner with respect to a global system or 2) it may be defined by specifying the location of one corner with respect to a global coordinate system and the dimensions of the cuboid. It is possible to derive one from the other.

**Definition of Adjacency between a space and outside.** An activity space is adjacent with respect to the outside if it is possible to define a rectangular cuboid that is adjacent to the boundary that does not intersect other physical objects in the design. Two dimensions of this cuboid will be determined by the dimensions of the activity space (or boundary) while the third dimension is typically at 5 feet as shown by the shaded rectangle in the Figure 27.

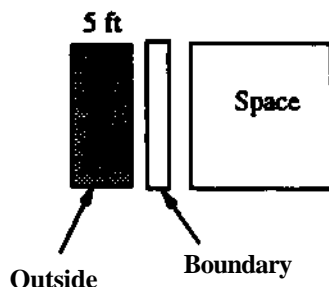


Figure 27: Definition of adjacency to outside

**Control.** The designer controls the process by deciding to focus on some subset (one or more) of the active objectives at any time. Before describing the process, the current state of the variables and objectives is described.

**Variable Types used in the example.** The following variable types are used in this example: 1) building, 2) activity-space, 3) boundary, and 4) opening. The building type is an aggregate type described by: 1) list of activity-spaces, 2) structural system, and, 3) HVAC system. There are other aspects (e.g., water supply and drainage systems) in a building but this example will deal primarily with the building spaces and consider their interaction with structural and HVAC systems.

The environment variable type may be described as an aggregation of types defining various aspects of the local environment. In this example, we are primarily concerned with the noise-source which may be represented as: 1) Location of point source (wrt some global coordinate system), and 2) Intensity of sound level in dB at some distance from the source.

The geometry for all entities is a rectangular cuboid located at some position in space (wrt to a defined site coordinate system).

An activity space is **an aggregate** type described by: 1) geometry, 2) boundary-list, 3) opening-list, and 4) **content-list**. The boundary list is the list of boundaries associated with the space (both external and **internal**). Boundaries may be shared by activity spaces. The content list is a list of objects that occur **within the** activity space, e.g., beams and columns (or portions of beams and columns) from structural systems, ducts from HVAC systems etc.

A boundary is **an aggregate** type described by: 1) geometry, 2) layer list, and 3) opening list. The layer list is a list of boundary layers that make up the boundary. The opening list is a list of openings that lie on the boundary.

A layer is an aggregate type described by: 1) geometry, and 2) material. The material is some building material (e.g., styrofoam) that constitutes the layer.

An opening is an aggregate type described by: 1) geometry, and 2) material. The material of an opening refers to the material used for glazing in case the opening is a-window or skylight or the material of the door in case the opening is a door.

**Variable Bindings and Objectives at start of process.** The space list for the building has been determined as shown below. Other variables are not yet bound. The spaces are represented by the variables: Kitchen, Dining, Living, Bed, Bath. Each of the variables Kitchen, Living, Bed, Bath, and Dining are of type activity space and are unbound. Some other variables include the external visual obstructions Vobs (in daylight objectives) and the Acoustic-obs (in sound level objective). In this example there are no external visual obstructions at this point. However, since parts of the design themselves may become potential visual obstructions - so the value for Vobs might change. A example is given for each of the classes of objectives:

reduce-conducted-heat-loss(Bldg, Bldg-env)  
 floor-area(L, A1), equal-tolerance(A1, 144, 10)  
 daylight-factor(L, Mid, Clear, Vobs, 2,3)  
 ambient-sound-level(A1, Acoustic-obs, N-source, 50)  
 circulation(L,B, 7, 3)

Also, validity constraints are introduced for each of the variables (activity spaces and building).

Refinement of conducted heat loss. The following objective is in focus:

reduce-conducted-heat-loss(Bldg, Bldg-env)

Two refinements are created for this objective:

1. reduce-boundary-area-exposed-to-outside(Bldg, Bldg-env)
2. reduce-U-value-for-outside-boundaries(Bldg, Bldg-env)

Conducted heat loss depends on the area of the boundary exposed to the outside, the U-value (coefficient of thermal conductance) of the layers of material that make up the boundary and the temperature difference between the inside and outside as shown below:

$$\text{Min } A_{hl} * U_{bl} * \delta(T_x) + \dots + A_{bn} * U_{bn} * \delta(T_n)$$

where  $AK$  is the area of the boundary exposed to outside,  $U_{bi}$  is its thermal conductance and  $\delta(T_i)$  is the temperature difference. Notice that this formulation is equivalent to the refinement expressed earlier only if the values of  $\delta(T_i)$  and  $U_n$  are the same (or almost same) for the different spaces.

**Refinement of sound level objective.** Two alternative strategies are possible for achieving sound isolation:

1. Alt 1:  $\text{disUace}(A1, N\text{-source}, D), D > \sqrt{D_s * D_s * (\text{antilog}((L_s - 160)/10) / \text{antilog}((L_{act} - 160)/10))}$
2. Alt 2:  $\text{blocks-path}(A1, N\text{-source}, A_{obs}), \text{lower-limit-sound-trans-loss}(A_{obs}, L_s - L_{act})$ .

An acoustic obstruction maybe an earth berm, other activity spaces, boundaries. At this point no decision is made to use any one of these options for an acoustic obstruction. For example, consider the objective:

ambient-sound-level(Bed, Aobs, N-source, 30)

The following refinements are generated:

1. distance(Bed, N-source, D),  $D > 3162277.6$
2. blocks-path(Bed, N-source, Aobs), lower-limit-sound-trans-loss(Aobs, 45)

In the case of this refinement, there are two choices possible. Alternative 1 is eliminated by evaluation with respect to the objective that requires all of activity spaces to lie within the site boundaries. The distance requirement is too large for the Bed space to lie within site boundaries. This evaluation is deterministic but must be performed at a qualitative level, i.e., the geometry of the space is not determined as of yet. Thus, alternative 2 is included in the current design.

**Refinement of daylight objectives.** The daylight objective is refined into the following sub-objectives:

- the space must be adjacent to the outside; and
- the daylight through all openings at the reference point in the space meet the required criteria.

It must be noted that there are ways to get daylight to the space without the space being adjacent to the outside, e.g., light funnels etc. However, these alternatives are not considered in this example. For example, consider the objective:

daylight-factor(Living, P, Vobs, 2, 3)

The following refinements are generated:

1. outside-adjacent(Living, Bldg-env, Blistl)
2. partof(Blistl, Living.boundary-list)
3. daylight-thro-openings(Olist, Living, Blistl, Vobs, 2, 3)
4. partof(Olist, Living.opening-list)

The partof() objective constrains the boundaries in Blistl to be part of the activity space boundary list and similarly for the openings. Similarly, all of the other daylight objectives are refined.

**Refinement of circulation objective.** Each objective requiring circulation between two spaces is refined into two sub-objectives: (1) the two spaces must be adjacent. (2) if there is a boundary there must be an opening to allow for passage, (3) the size of the opening is greater than the dimensions and (4) the opening lies at the floor level of both the spaces. For example, consider the objective:

circulation!Kitchen. Dining, 7, 3)

The following refinements are generated:

1. side-adjacent(Kitchen, Dining, B6, 7, 3)
2. member(B6, Kitchen.boundary-list)
3. member(B6<sub>1</sub>, Dining.boundary-list)
4. opening(Okd, B6), Okd.height > 7, Okd.width > 3, opening-at-floor-level(Okd, Kitchen.geometry, Dining.geometry)
5. member(Okd, Kitchen.opening-list), member(Okd, Dining.opening-list)

"Okd" is a variable of type opening that represents the opening between the Kitchen and Dining spaces. The side-adjacent() objective constrains two spaces to have a common vertical boundary with certain minimum dimensions (in the above case 7 feet and 3 feet).

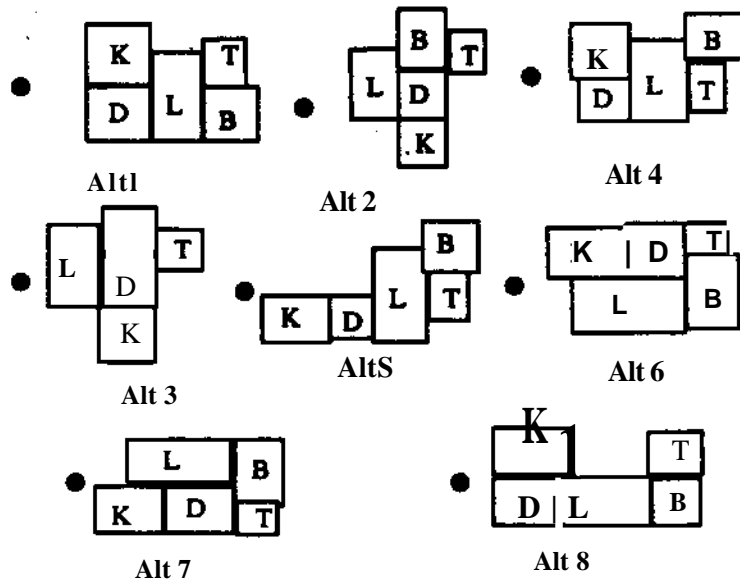
**Generation of Spatial layouts.** Now the designer brings into focus all spatial (perhaps geometry is a more appropriate term) objectives and tries to generate spatial layouts. Thus the following objectives are brought into focus:

1. floor-area() objectives;
2. outside-adjacent() objectives generated by refining daylight;
3. side-adjacent() objectives generated by refining circulation;
4. blocks-path() generated by refining sound level; and
5. validity constraints.

The designer can now add some new objectives specifying the minimum values for lengths and breadths of the spaces, e.g., lengths of all spaces except the bath must be at least 10 feet. In addition, the lengths and breadths of all the spaces are assumed:

1. Kitchen: 10 x 12 ft
2. Living: 10 X 15 feet
3. Bed: 10 x 13 feet
4. Dining: 10 x 10 ft
5. Bath: 6 x 8 feet.

**Also,** the **height** of all spaces is assumed to be 10 feet. All of these are recorded as assumption\*. These assumptions are common to all of the alternatives that will be generated. Besides the assumptions there may be assumptions associated with each alternative, i.e., alternative specific assumptions. The alternatives generated in this example are shown in Figure 28. The set of alternatives described here is not exhaustive; a number of implicit assumptions are made in generating the alternatives. Generating the alternatives automatically is not trivial. Tools such as ABLOOS [5] may be used to do this. However going from the description of the layout problem given in th»



Dot indicates noise source for reference (not to scale)

Figure 28: Alternative spatial layouts

form of objectives to ABLOOS is a fairly involved process. But it is still only a translation problem. A more difficult problem is to identify (automatically) the appropriate operator and ensure that the objectives and assumptions are sufficient to define one or more solutions. The emphasis of this research is not to automate the solution of the design problems but to identify and record any information generated and used in the process of solution. In the case of this example, ABLOOS was not used.

**Evaluation and selection of layout alternatives.** Objectives in focus for evaluation:

1.  $O_1$ : reduce-boundary-area-exposed-to-outside(Bldg, B-env)
2.  $O_2$ : like-shape(Bldg)

The first objective involving the boundary area is a reduce type of objective, the evaluation return\* a value for the characteristic: boundary area exposed to the outside. This characteristic is measured on a ratio scale and in square feet units.

The second objective is not defined. This may occur because the designer is not able to articulate a definition but is still able to evaluate the alternatives with respect to the objective. In the case of the second objective the designer assigns a ranking from 0 to 8 (larger value means better).



| Alt or Obj | $O_1$ | $O_2$ |
|------------|-------|-------|
| $A_1$      | 1100  | 0     |
| $A^*$      | 1220  | 0     |
| $A_2$      | 1220  | 0     |
| $A_4$      | 1040  | 0     |
| $A_5$      | 1260  | 0     |
| $M$        | 1040  | 1     |
| $A_7$      | 1040  | 0     |
| $A_8$      | 1160  | 0     |

Table 7: Matrix showing results of evaluation

it is better; there is no notion of how much better) for each of the alternatives. In this case the characteristic is being measured on an ordinal scale. The evaluation matrix is shown in Figure 7. In this case it is seen that alternative  $AQ$  will be selected as it is preferred by both objectives. If instead that the designer had picked  $A3$  as the most preferred alternative with respect to  $O_0$ , there would be a conflict because  $A3$  is not one of alternatives preferred by  $O_1$ . Typically, such a conflict would be resolved by the designer stating that  $O_0$  is more important objective than  $O_1$ .

The selected alternative is shown in Figure 29. After the selection of this alternative the

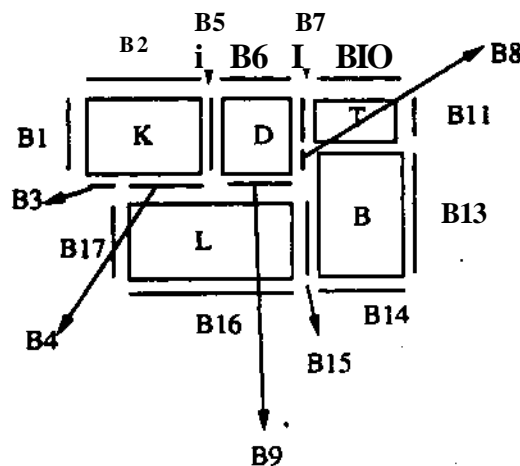


Figure 29: Chosen alternative with boundaries

current state of the variable bindings changes for all the activity spaces as shown below for the Kitchen:



**Kitchen**

- geometry -> 50, 60, 10, 60, 72, 20
- boundary-list — B1, B2, B5, B3, B4
- opening-list —• Okd
- content-list —• [..]

Figure 30 shows a pictorial representation of the design process up to this point. All of the objectives, operators and validity constraints are not shown in the Figure 30.

**Acoustic Obstructions.** The Kitchen, Dining and Living spaces serve as acoustic obstruc-

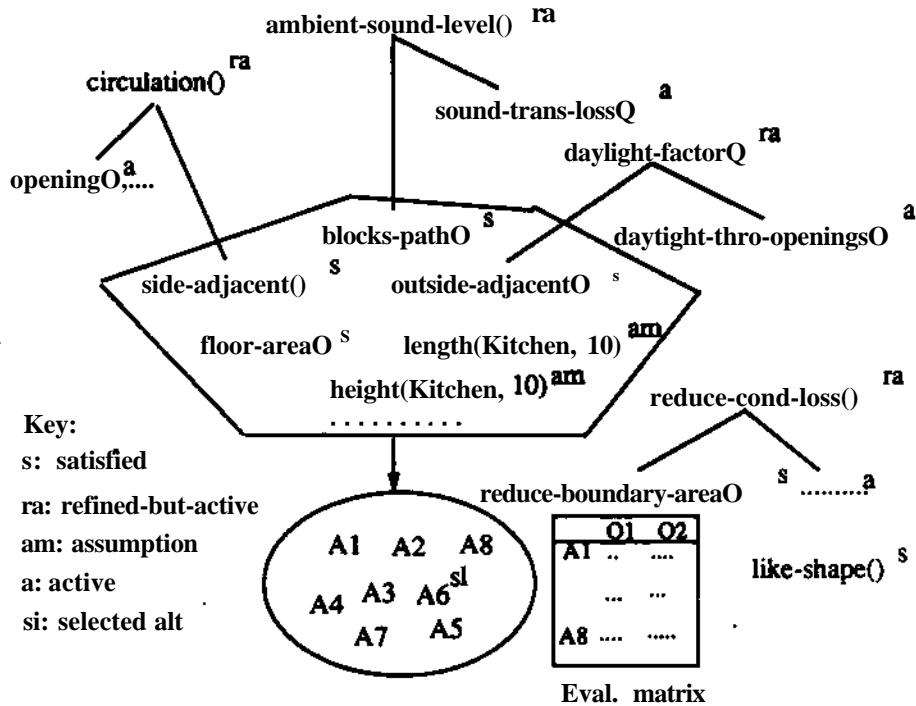


Figure 30: Pictorial representation of the design process

tions for the Bed space. Figure 31 shows the two possible sound transmission paths from the noise source through the obstructions to the Bed space: 1) B1, B5, B8, and 2) B17, B18. Corresponding to these two paths the *sound-transmission-loss()* objective implies that the total loss:

$$B1 + B5 + B8 = \mathbf{B11} + \mathbf{B15} = 45 \text{ dB.}$$

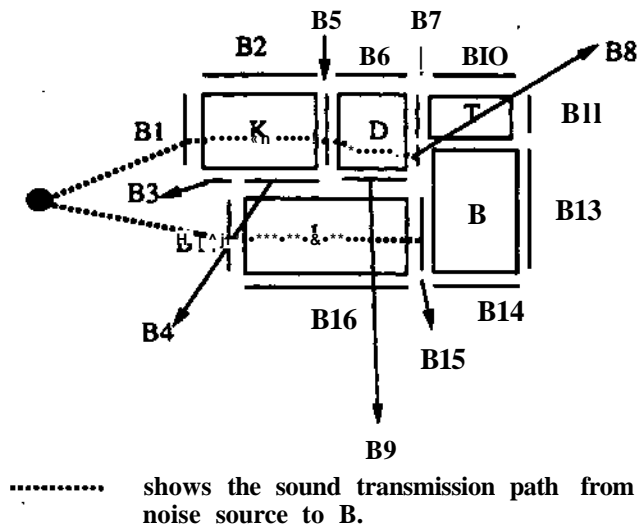


Figure 31: Sound transmission paths

The loss occurring over the distance of the spaces themselves is ignored. If the loss is to be distributed uniformly, it implies that the boundaries B1, B5, and B8 must each have an STL of at least 15 dB. while, B17 and B15 must each have an STL of at least 22.5 dB.

Designing for daylight in living space. The following objective is in focus:

daylight-thro-openings(Living, Blistl,Vobs, 2, 3)

This problem is very underconstrained. We could have more than one opening to satisfy the requirement. Moreover, there are two boundaries (B16, B17) on which the openings can be placed. At the opening may be placed anywhere on the boundary. In dimensioning the opening assumption\* have to be made regarding the dirt factor (which is a reflection on the maintenance), the allowance for mullions, and the type of glazing to be used. These factors affect the amount of daylight entering through an opening. Also, the internally reflected contribution to the daylight within the space depends on the reflectances of the interior boundaries. The designer makes the following assumptions:

- Use a single opening.
- Place the opening on B16.
- The centerline of the opening must be centered on the boundary.
- Allowance for dirt = 0.1).

- **Allowance for mullions** = 0.8.
- Glazing is clear single (transmission factor is 1.0).
- Reflectance of walls = 40 percent.
- Reflectance of floor = 20 percent.

The design requires a DF ~ 2 at the reference point which is 7.5 feet away from the boundary B16 and at the 7.5 from the boundary B17. As a result of the allowances, the design DF is  $2/(0.9*0.8) = 2.8$ . In the absence of any external obstructions,  $DF = S.C. + I.R.C.$ , i.e., the sum of the sky component and the internally reflected component. The internally reflected component depends on the reflectances of the walls and floor.

Even with all the assumptions, there are a variety of window sizes (height vs width) that will satisfy the objective. These sizes can be arrived at by trial and error. For example, suppose that the required DF of 2.8 will be provided by a S.C. of 2.3 and a I.R.C of 0.5. Since the window must be located centered on the boundary, the reference point falls at the center of the opening. So we will consider two hypothetical openings that together define the whole opening. See Figure 32. Each of

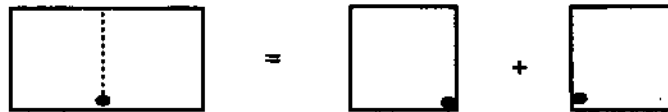
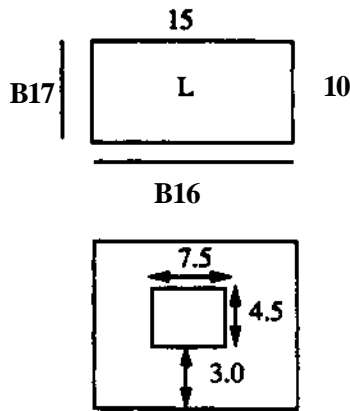


Figure 32: Two Hypothetical Openings

these openings provide  $1/2$  of  $2.3 = 1.15$  of DF. From the table for computing the sky-component (pg. 202 of [15]) :  $H/D = 0.6$  and  $VV/D = 0.5$  provide an S.C. = 1.2. Since  $D = 7.5$ , the values of  $H = 4.5$  ft and  $W = 3.75$  ft. Thus total area of opening =  $4.5 \times (3.75 \times 2) = 33.75$ . From another table for computing the internal reflected component (pg. 184 of [15]), we can determine the I.R.C based on the ratio of opening area to floor area and the reflectances. This value turns out to be  $0.6$ . Thus, we have a  $DF = 2 \times 1.2 + 0.6 = 3.0$ . This is acceptable and hence the sizes of the opening are as shown in Figure 33. This is only one of a number of possible sizes for the opening.

This refinement **has** generated bindings for the geometry and material of the opening in the living space to **meet the** daylight requirements. For the daylight requirement to be satisfied we must ensure that there be **SO** visual obstructions in the living space between the opening and the reference point **as shown by triangular** (in plan) volume ABC in Figure 34. This could be handled by establishing the objective: **no-visual-obstructions**(Living, OLiving, mid). Since the current contents of the living space are empty, this objective is currently satisfied. However, should the contents of the space change as other objectives (e.g.. structural or HVAC related) are achieved then this objective roughly be violated.



All dimensions in feet

Figure 33: Dimensions of opening for daylight in Living

We have shown snippets of the design process for the residential building. A number of objectives remain to be satisfied. The process is documented in the design record. This documentation is useful to "explain" design decisions and track the effect of modifying decisions.

## 7 Summary

The goals of this research are to: 1) identify the components of design intent, i.e., the kinds of information we need to capture to be able to support the tasks of explanation, modification and reuse; and, 2) to identify computable representations for the different components.

Our approach to this problem develops a model of design decision-making that explicit

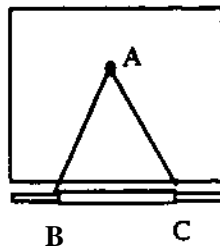


Figure 34: No obstruction requirement

identifies intent. There are two characteristics that distinguish our approach to design: 1) objectives are used to define the design problem and represent intermediate stages leading up to the final solution, and, 2) the process is systematic involving the generation and evaluation of alternatives at every stage. An objective is a requirement that the design artifact must satisfy (functional, cost, aesthetic, constructability etc). Computationally, objectives are relationships over variables. Variables may represent given information on climate, cost etc (exogenous variables) or they may represent aspects of the design (design variables). Design variables may represent entities or characteristics. An entity is an object that is ultimately described in terms of geometry and/or material. A design characteristic is some property measured over the entities (e.g. floor area, exposed boundary area etc.). Entities may be related in aggregation or specialization hierarchies. Importance can be associated with objectives either as an ordinal ranking or in the form of weights.

The design problem is described as a conjunction of objectives. The initial objectives need not be complete nor consistent. Also, existing objectives may be modified during the design process and new objectives may be introduced at any point in the process. The design process model is characterized by five activities: focus, refine, evaluate, select and resolve. The objectives can be addressed in some order by focussing on a subset of them. Objectives are achieved by a process of refinement which may result in a set of alternative bindings for the design variables involved, or, in alternative decompositions for the objectives. Assumptions made in the generation of the alternatives are recorded. We assume the existence of operators (refinement operators) for the generation of alternatives. These operators describe all possible alternatives for achieving some objective. In a particular design situation one or more of these alternatives may not be applicable. This is determined in the process of evaluation where alternatives are evaluated with respect to other objectives. When reduce type of objectives are involved the performance of alternatives with respect to these objectives are compared and the "best" alternative is selected. The criteria used for selecting the best alternative depends on the way in which the importance is expressed. The design is completed when the current set of objectives is satisfied.

The history of the design process is recorded in a representation called the design record. The record can be used to support the tasks of explaining why certain decisions were made and identifying the ramifications of modifying certain decisions. The different components of intent in this model are: 1) objectives, 2) importance associated with groups of objectives, 3) alternatives generated by refinement, 4) assumptions made in the refinement; 5) the evaluation and selection from these alternatives, and, 6) focus sequence in which the objectives are addressed.

We are developing an interactive computer-based environment to demonstrate this approach. In this environment the designer is able to 1) define a design problem by instantiating objectives (new-objectives may be introduced at any time), 2) solve the problem interactively, 3) maintain a record of the solution process, and, 4) use this record to seek explanations or during a resolve situation as explained earlier. Our application domain is the design of residential buildings. At the present time, we use a constraint logic language, CLP(Tfc) [9] to express objectives and operators (this is subject to change). CLP(TJ) is a constraint logic language, which combines logic programming with an incremental linear equation solver within a general scheme known as constraint logic programming (CLP) [8]. An interface is being built in X using the Motif widget set [30].

The following are limitations or research issues or just some general concerns related to this research:

- **A piece of information that is missing in the current model** is the reasoning for why the objectives are addressed in some sequence. Some objectives may be addressed before others because of information dependencies or for reasons of efficiency.
- **We have begun** to identify a classification for objectives but it needs to be completed. There are some classes that are domain independent, i.e., at least applicable over a number of domains.
- **It** may not always be possible to find computable representations for objectives. Designers may find it difficult to articulate their objective although they are able to use it to evaluate design alternatives. Although we allow for informal textual descriptions of objectives, they are interpreted by the designer and are treated separate from computable representations. We need to look at the integration of informal and the computable representations.
- **How** practical is such an approach? Many researchers in various domains have prescribed systematic approaches to design [12], [17], [25]. It has also been noted that most designers do not follow a systematic approach [23], [25]. This question can be answered only by usability studies conducted within design offices. We believe that the acceptability of this approach will depend on the availability of a large vocabulary of objectives and operators to generate alternatives for the different classes of objectives in a design domain. In this regard, we expect that through continuous use of this paradigm, a library of objectives, operators and types of design variables, will be created.
- **Learning:** It may be possible for the user to go over the record of the solution process a posteriori and identify places where wrong decisions were made and identify rules (or heuristics) that might at the very least improve the process for the same design problem. This is something to be considered for future research.

This research introduces a different approach to design that focusses on the objectives instead of the artifact or its components. The approach raises a lot of issues and it is not possible to completely address all of them within the scope of this work.

## References

- [1] Hong-Tai Chou and Won Kim. Versions and change notification in an object-oriented database system. In *Proceedings of 25th ACM/IEEE Design Automation Conference*, pages 275-281. IEEE, 1988.
- [2] Hwang C.L. and K. Yoon. *Multiple Attribute Decision Making : a state-of-the-art-surrey*. Springer-Verlag, New York, 1981.
- [3] Jeff Conklin and Michael L. Begeman. gIBIS: A hypertext tool for exploratory policy discussion. *ACM Transactions on Office Information Systems*, 6(4):303-331, 1988.
- [4] M. David Egan. *Architectural Acoustics*. McGraw Hill, New York, 1988.

- [5] U. Flemming, R. Coyne, T. Glavin, and M. Rychener. *A Generative Expert System for the Design of Building Layouts - Version 2*, pages 445-464. Computational Mechanics Publications, Southampton, 1988.
- [6] A.C.B. Garcia and H.C. Howard. Building a model for augmented design documentation. In *Proceedings of the First International Conference on Artificial Intelligence in Design*, Edinburgh, June 25-27 1991.
- [7] James Garrett. *A Standards Processing Expert System*. PhD thesis, Department of Civil Engineering, Carnegie-Mellon University, May 1985.
- [8] J. Jaffar and J-L. Lassez. Constraint logic programming. In *Proceedings of the 14th ACM Principles of Programming Languages Conference*, 1987.
- [9] Joxan Jaffar, Spiro Michaylov, Peter J. Stuckey, and Roland H. C. Yap. The clp(r) language and system. Report CMU-CS-90-181, Department of Computer Science, Carnegie-Mellon, Pittsburgh, PA, 1990.
- [10] W. Kim, Jay Banerjee, H. Chou, and J.F. Garza. Object-oriented database support for CAD. *Computer-Aided Design*, 22(8), 1990.
- [11] J.L. Kolodner, R.L. Simpson, and K. Sycara. A process model of case-based reasoning in problem solving. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 284 - 290, 1985.
- [12] T.Y. Lin and S.D. Stotesbury. *Structural Concepts and Systems for Architects and Engineers*. Wiley, New York, 1981.
- [13] R. J. McCall, G. Fischer, and A. Morch. *Supporting Reflection-in-Action in the JANUS Design Environment*, pages 247-259. The MIT Press, Cambridge, Mass., 1990.
- [14] T.M. Mitchell, L.I. Steinberg, and S. Kedar-Cabelli. An intelligent aid for circuit redesign. In *Proceedings of National Conference on Artificial Intelligence, AAAI-83*, 1983.
- [15] Fuller Moore. *Concepts and Practice of Architectural Daylighting*. Van Nostrand Reinhold, New-York, 1985.
- [16] J. Mostow. Design by derivational analogy: Issues in the automated replay of design plans. Technical report, Department of Computer Science, Rutgers University, New Brunswick, NJ. 1987.
- [17] Gerhard Pahl and Wolfgang Beitz. *Engineering Design*. The Design Council, Springer-Verlag\*. London, 1984.
- [18] Colin Potts and Glenn Burns. Recording the reasons for design decisions. In *Proceedings of 10th International Conference on Software Engineering*, 1988.
- [19] Daniel Rehak. By personal communication, 1990.



- [20] H. Rittel **and** M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, (4), 1973.
- [21] J.R. Rossignac, P. Borrel, and L.R. Nackman. *Interactive Design with Sequences of Parameterized Transformations*. Eurographics Seminar Book. Springer-Verlag, New York, 1988.
- [22] T. L. Saaty. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15(2):234-281, 1977.
- [23] Larry A. Stauffer, David G. Ullman, and Thomas G. Dietterich. Protocol analysis of mechanical engineering design. In W. E. Eder, editor, *Proceedings of the 1987 International Conference on Engineering Design, ICED 87*, pages 74-85, Cambridge, MA, August 1987. American Society of Mechanical Engineers.
- [24] S.S. Stevens. *Measurement, psychophysics, and utility*, pages 18-63. John Wiley, New York, 1959.
- [25] Fred A. Stitt. *Designing Buildings That Work*. McGraw-Hill, New York, 1985.
- [26] W. Swartout. A system for creating and explaining expert consulting systems. *Artificial Intelligence*, 21(3), 1983.
- [27] J.B. Thompson and S. Lu. Design evolution management: A methodology for representing and using design rationale. In *Proceedings of the Second International ASME Conference on Design Theory and Methodology*, 1990.
- [28] V.H. Hartkopf, V.E. Loftness, and P.A.D. Mill. *The Concept of Total Building Performance and Building Diagnostics*. Special Technical Publication 901, ASTM, Philadelphia, PA, 1988.
- [29] D. Wile. Program developments: Formal explanations of implementations. *Communications of the ACM*, 26(11), 1980.
- [30] Douglas A. Young. *The X Window System: Programming and Applications with Xt*. Prentice-Hall, New Jersey, 1990.