22. Reading Architectural Plans: A Computable Model

Sally McLaughlin

Department of Architectural and Design Science
University of Sydney
NSW 2006, Australia

A fundamental aspect of the expertise of the architectural designer is the ability to assess the quality of existing and developing proposals from schematic representations such as plans, elevations and sections. In this paper I present a computable model of those aspects of the evaluation of architectural floor plans that I believe to be ammenable to rule-like formulation. The objective behind the development of this model is twofold: 1) to articulate a belief about the role of simple symbolic representations in the task of evaluation, a task which relies primarily on uniquely human capabilities; and 2) to explore the possible uses of such representations in the development of design expertise.

Input to the model consists of a specification of a design proposal in terms of walls, doors, windows, openings and spaces together with a specification of the context in which the proposal has been developed. Information about context is used to retrieve the goal packages relevant to the evaluation of the proposal. The goal packages encode information about requirements such as circulation, visual privacy and thermal performance. Generic associations between aspects of a plan and individual goals are used to establish if and how each of the goals have been achieved in the given proposal. These associations formalize relationships between aspects of the topology of the artefact, such as the existence of a door between two rooms, and a goal, in this case the goal of achieving circulation between those two rooms. Output from the model consists of both a graphic representation of the way in which goals are achieved and a list of those goals that have not been achieved. The list of goals not achieved provides a means of accessing appropriate design recommendations. What the model provides is essentially a computational tool for exploring the value judgements made in a particular proposal given a set of predefined requirements such as those to be found in design recommendation literature.

Introduction

A fundamental aspect of the expertise of the architectural designer is the ability to assess the quality of existing and developing proposals. The knowledge intensive nature of this exercise is evident when we consider the nature of the kinds of comments that may be elicited during the course of an examination of a residential floor plan: comments such as "this room will be open and sunny"; "I don't like the relationship between the kitchen and dining room"; or "I don't think having the entry here is going to work"? How is the designer to know that a particular room will be open or sunny? What information allows him to recognize a good or bad relationship between kitchen and dining areas? Finally, how is he able to focus upon particular aspects of the plan and to recognize the contribution that they make to the overall quality of the proposal?

Whilst the ability to perform such evaluations is integrally linked to the designer's ability to create rich, personal simulations of a proposal (Foz 1973), there is a sense in which rule-like expressions may be employed. Consider the following information extracted from a summary of research relating to the quality of planning in family housing (Zeisel and Welch 1981):

When living rooms are used formally by families, they tend to dislike having the main unit entry directly into that formal area. To them it means unnecessarily invading this place in the home.

A number of points may be made in relation to this passage that I believe apply to design recommendation literature in general. First, the passage indicates a particular configuration to be avoided, an entry opening directly into a formal living area, but it is not prescriptive, there are many configurations that would satisfy this requirement. Second, an attempt has been made to convey something of the perspective from which the recommendation is being made, the configuration is disliked because residents feel it allows unnecessary intrusion into formal living areas. This kind of information is important in constructing a perspective from which the rich simulations necessary to the evaluation of an architectural proposal may take place. Finally, the recommendation implicit in the passage should not necessarily be universally applied. The appropriateness and priority given to the considerations expressed have to be assessed in the context of individual proposals, there may be situations where considerations such as a tightly constrained site or the desire to create an immediate feeling of spatiousness upon entering may override the privacy issue.

In this paper I present a symbolic representation of rule-like information extracted in part from design recommendation literature. A computable model is developed that supports the application of these "rules" whilst attempting to respect the nature and limitations of design recommendations as outlined above. The model consists of two components: first, a database of the packages of goals that a designer may bring to bear in the assessment of a proposal in a given context; and second, generic associations between aspects of the plan representation and goals. The objective in developing the model was to apply the rules based on design recommendation literature to individual proposals in such a way as to expose the tradeoffs, the value judgements, that have been made in the course of developing those proposals. In addition, a means of conveniently accessing recommendations so as to fill out the perspective from which the appropriateness of these decisions may be determined, was sought.

Goal Packages

It is well recognized that a designer exploits his knowledge of building types as a basis for proceeding with the development of a design proposal (Foz 1973, Darke 1979, Hillier et al. 1972). Similarly the divisions that occur in design recommendation and design code literature indicate that knowledge of building types is important in the evaluation of design proposals. For example, it is not uncommon for planning information to be organized under categories such as family housing, housing for the elderly, semi-detached housing, home units, office buildings, and buildings for various categories of environmental conditions. In developing a representation of design "rules" based on an interpretation of the design recommendation literature discussed here, I found it useful organize information into sets of constraints indexed under such categories. A group of constraints indexed under a particular context description will be referred to as a goal package. The goal package indexed under the context description *building type = residential, occupants = family* is listed in Table 1. The goal package for the context *goal = optimize_thermal_environment, climate = temperate* is listed in Table 2. The principle advantage of grouping constraints into packages is that it allowed many assumptions about the nature of the artefact to which the constraints would be applied to be made.

---

entry(X)
living(X)
family(X)
dining(X)
# Circulation is required between the entry and the main access to the house. The route will be used by the residents and both formal and informal visitors. #
goal = circ_reln(X,Y, (residents,formal_visitors,informal_visitors))
    where entry(X) and outdoors(Y) and primary_access_from(Y)
goal = circ_reln(X,Y,residents)
    where outdoors(X) and access_from(X) and
    for all Y where store(Y) and contains(outdoor_equipment,Y)
# Circulation is required between the entry and informal areas. #
goal = circ_reln(X,Y,(residents,informal_visitors))
    where entry(X) and
    for all Y where room(Y) and part_of(zone_informal_living,Y)
goal = circ_reln(X,Y,(residents,formal_visitors,informal_visitors))
    where entry(X) and living(Y)
goal = circ_reln(X,Y, (residents,formal_visitors,informal_visitors))
    where living(X) and dining(Y)
goal = circ_reln(X,Y, (residents,formal_visitors,informal_visitors))
    where bathroom(X) and for all Y where [living(Y) or dining(Y)]
goal = circ_reln(X,Y,residents)
    where bathroom(X) and for all Y where bedroom(Y)
goal = circ_reln(X,Y,residents)
    where laundry(X) and for all Y where outdoors(Y) and
    [contains(outdoor_clothes_drying,Y) or contains(cleaning_equipment,Y)]
goal = circ_reln(X,Y,(residents,informal_visitors))
    where kitchen(X) and for all Y where outdoors(Y) and
    [contains(outdoor_play,Y) or contains(cleaning_equipment,Y)]
goal = visual_barrier(X,Y)
    where entry(X) and for all Y where room(Y)
goal = visual_barrier(X,Y)
    for all X where store(X) and for all Y where room(Y)
goal = visual_access(X,Y)
    where kitchen(X) and for all Y where outdoors(Y) and contains(outdoor_play,Y)

# Informal parts of the house should not be able to be viewed by formal visitors #
goal = visual_barrier(X,Y)
    where room(X) and room(Y) and part_of(zone_informal,X)
    and for all Y on_circ_path(formal_visitors,Y)
# Normal activities of residents should not be able to be viewed by formal visitors #
goal = visual_barrier(X,Y)
    where room(X) and room(Y) and part_of(zone_formal,X) and
    for all Y on_circ_path(residents,Y)

Table 1. Goal package for context: building type = residential, occupants = family. This information has been developed primarily by interpreting the recommendations outlined in Zeisel and Welch [1981].

# Minimize heat loss to outdoors and heat gain in summer #
goal = thermal_barrier(X,Y) for all X where room(X) and for all Y where outdoors(Y)
# Minimize heat loss between rooms. #
goal = thermal_barrier(X,Y) for all room(X) and heated (X) and for all room(Y) and !heated(Y)
goal = sunny(X,winter) for all X where room(X) and part_of(zone_living,X)
goal = sunny(X,winter) for all X where bedroom(X)
# not high priority - bedrooms don't need to be as warm as living areas #
goal = not_sunny(X,summer) for all room(X)
goal = wind_barrier(X,Y) for all X where room(X) and for all Y where winter_wind_source(Y)
goal = wind_access(X,Y) for all X where summer_wind_source(X) and direction(Y) and
    opposite(X,Y)
goal = construction_type(X,Y) where heavy_weight(Y) or insulated_frame(Y)
        for all X where room(X)

Table 2. Goal package for context: goal = optimize_thermal_environment, climate = temperate. Information derived from an interpretation of NSB 21 [1971] *House design for temperate climates*.

---

part_of(zone_informal_living,X) :- bedroom(X) or kitchen(X) or family(X)
part_of(zone_informal,X) :- bedroom(X) or kitchen(X) or family(X) or laundry(X) or store(X)
part_of(zone_formal,X) :- living(X) or dining(X)
on_circ_path(X,Y) :- variable_member(Proof(circ_reln(A,B,L)),Y) and member(X,L)
part_of(zone_living,X) :- living(X) or family(X) or dining(X) or kitchen(X)
part_of(zone_winter_living,X) :- kitchen(X) or dining(X) or winter_living(X) or winter_family(X)
summer_wind_source(X) :- location = Sydney and [X = NE or X = E]
winter_wind_source(X) :- location = Sydney and [X = NW or X = W or X = SW or X = S]
opaque(X) :- room(X) or wall(X) or door(X)
window(X) :- glazed_window(X)
object(X) :- room(X) or wall(X) or door(X) or window(X)

Table 3. Background knowledge required to interpret goal packages in Tables 1-4. This information is stored in frames.

In this section I wish to consider the relationship between some of the constraints contained in the goal package listed in Table 1 and the design recommendation literature on which they were based. Consider the following extract from Zeisel and Welch [1981] in their summary of research relating to the design of housing for families:

In their home, residents tend to designate separate places for more formal behaviour such as entertaining and showing off special furniture or a console TV, and places designated for informal activity such as eating breakfast, children playing, and sitting around when a casual friend drops in. Formal areas may usefully be thought of as "front stage" areas where residents present their "best side" to others; informal areas as "back stage" where residents let down their hair.

The recommendations suggest a need for formal and informal living areas. This is represented in the goal package by proposing that there should be three separate living areas, *living(X), family(X)* and *dining(X)* and by associating individual rooms with the zones *zone_informal*, *zone_informal_living* and *zone_formal* (see Table 3). The effect of asserting the existence of specific room types within the goal package is that if rooms of these types cannot be found in the design proposal to which the goal package is being applied, a message about the absence of such a room in the proposal will be posted. For example, if the goal package listed in Table 1 were applied to a plan not containing a room of type *living* then the violated constraint *living(X)* together with the appropriate design recommendation would be displayed:

Housing evaluation studies show that many residents prefer to have a living area in their homes that they can use as a "showcase living room" separable from other areas of the dwelling unit, especially where children play.

The effect of associating room types with particular zones is evident when we consider the representation of recommendations such as the following:

The access to living rooms should by-pass dining/den so visitors would not have to pass through a messy all-purpose room.

This recommendation is generalized in the goal package to incorporate the idea that none of the informal parts of the house should be able to be viewed by formal visitors. This objective is represented as follows:

goal = visual_barrier(X,Y)
    where room(X) and room(Y) and part_of(zone_informal,X)
    and for all Y on_circ_path(formal_visitors,Y).

The predicate *on_circ_path(formal_visitors,X)* (see Table 3) is established by identifying all those rooms that must be passed through by formal visitors in their normal patterns of circulation. The normal patterns of circulation of formal visitors is established by reference to the following constraints:

goal = circ_reln(X,Y, (residents,formal_visitors,informal_visitors))
    where entry(X) and outdoors(Y) and primary_access_from(Y)
goal = circ_reln(X,Y,(residents,formal_visitors,informal_visitors))
    where entry(X) and living(Y)
goal = circ_reln(X,Y, (residents,formal_visitors,informal_visitors))
    where living(X) and dining(Y)
goal = circ_reln(X,Y, (residents,formal_visitors,informal_visitors))
    where bathroom(X) and for all Y where [living(Y) or dining(Y)].

The first statement represents the fact that circulation is required between the outdoor area that is the principle access to the house and the entry. The second statement conveys the fact that circulation is required between the entry and living room and so on. The effect of defining a *circ_reln* is to specify that two rooms should be fairly close to each other, the goal will not be considered to be achieved if circulation from one room to the next cannot be achieved by passing through only a limited number of rooms. In the examples discussed in this paper this limit will be set to three. Returning to the constraint representing the objective that none of the informal parts of the house should be able to be viewed by formal visitors, this objective will be considered to be violated if there is not a visual barrier between each room associated with zone_informal and any room on the circulation path of formal visitors, ie. the circulation path between outdoors and entry, entry and living, living and dining and dining/living and a bathroom.

Representation of Artefact and Context Information

An important objective in developing a representational scheme for the floor plans has been to limit the assumptions made about the nature of relevant features at the time of inputting a representation of the artefact. This objective has to be balanced with that of providing a means of supporting simple and unambiguous input. The floor plans employed are represented in terms of walls, doors, windows, openings and spaces. Rooms are a special type of space. Various restrictions are imposed in relation to these entities at the time of input: doors and windows must be inserted into a wall; openings must span from one wall to another; spaces are defined by nominating the walls, doors, windows and openings that bound the space, and these elements must define a closed space. Information about artefacts is stored in frames.

A number of higher level features can be determined from these low level features. For example, the relation next_to(X,Y) can be determined for two spaces X and Y by determining if these spaces share either a door, window, wall or opening. These higher level features are determined as required. Other features may be determined by asking the user to nominate relevant information. Procedures for aquiring additional and higher level features are associated with *if-needed* slots in the relevant frames.

Context information is represented as lists of feature-value pairs. For example, the fact that the building is located in a temperate climate is represented by the feature-value pair *climate = temperate*.

Abstracting Aspects of an Artefact Relevant to the Current Context

In this section I will introduce a set of qualitative associations between the goal predicates used to define the goal packages and the predicates used to describe the topology of a floor plan. The purpose of these associations is to allow the means by which individual goals are achieved within a specific floor plan to be identified. The associations embody knowledge such as the observation that, in the Southern hemisphere, North facing windows create the possibility that a room will be sunny in winter.

The associations are in the form of rules describing the state of the environment that can be achieved given a specific topological configuration and an initial state of the environment.

Rules are selected by working back from the goal specification as follows: first, expanding the goal specification into an initial and final state specification; second, selecting a rule that will reduce the difference between those states; third, attempting to instantiate that rule, identifying the topological features potentially relevant to achieving the goal in the process; and finally determining whether the application of the rule did in fact achieve the final state from the initial state. If the final state has not been achieved an intermediate state is proposed and breadth-first backward chaining is employed.

Before exploring this process in more detail I wish to discuss the nature of the goal specification conversion to an initial and final state, and the form of the rules employed. The information required to convert the goal predicate *circ_reln* into an initial and final state specification is given in Table 4. This information is indexed in a goal conversion table under the index *circ_reln*. The information should be interpreted as the conversion of the requirement that *circulation be provided between spaces X and Y* to a requirement that *if in the initial state there is a person in space X*, that is the circulation pressure in X is positive, *it should be possible to achieve a final state where there is a person in space Y*. While the introduction of the notion of circulation pressure, or *circ_pressure*, is somewhat cumbersome it eliminates the need for recursive rule definitions.

Goal:
  circ_reln(X,Y,_)
Conversion:
  Initial_State: eq(circ_pressure(X),+)
  Final_State: eq(circ_pressure(Y),+)

Goal:
  visual_access(X,Y)
Conversion:
  Initial_State: eq(vis_pressure(X),+)
  Final_State: eq(vis_pressure(Y,+)

Goal:
  visual_barrier(X,Y)
Conversion:
  Initial_State: eq(vis_pressure(X),+)
  Final_State: eq(vis_pressure(Y),-)

Table 4. Goal conversion for predicates *circ_reln*, *visual_access* and *visual_barrier*.

The form of the rules is derived from that of the rules employed by Forbus [1984] to accomplish qualitative reasoning in the domain of physics. The names of some of the categories have changed and the *influences* section of the original rules has been dropped. The rule connecting the topological feature of adjacent spaces with an opening between them to the facilitation of allowing a person to move from one space to the other is given in Table 5. The first part of the rule, the *individuals* section, specifies the type of the topological features relevant to the rule. In this case variables A and B must be spaces and variable X a door or opening. The second section, the *topological relations* section, specifies the relations between those features that must hold if the rule is to be applied. In this case spaces A and B must be adjacent to each other, and a single door or opening X must be in both spaces. The separation of *individuals* and *topological relations* is necessary for efficient identification of appropriate instantiations of the rule. The topological relations significantly restrict the number of instantiations that need to be considered. The information in the *individuals* section is applied only after potential instantiations have been identified, as a final check on type. The *conditions* section specifies the value of state variables that must exist if the rule is to be applied. In this case the circulation pressure at A must be greater than the circulation pressure at B. This condition is necessary to ensure that rule sequences are constructed in an appropriate order. The *direction* section specifies the way in which the values of state variables will change as a result of the application of the rule. In this case application of the rule will cause a decrease in *circ_pressure* at space A and an increase in *circ_pressure* at space B. In other words, the rule is essentially a simulation of passage from space A to space B. The values 0, + and - are qualitative values: 0 indicates a steady state, + an increase in the value of a variable, and - a decrease in the value of a variable. The rule is indexed in the rule table under the index *circ_pressure*. Each rule is also indexed under a direction predicate. This is to ensure that the construction of a rule sequence proceeds in a single direction.

```
Direction:
   eq(circ_pressure(B),+)                      # circ_pressure(B) increasing #
Rule:
   Individuals:
        space(A)
        space(B)
        door(X) or opening(X)
   Topological Relations:
        next_to(A,B)
        in(A,X)
        in(B,X)
   Conditions:
        gt(circ_pressure(B),circ_pressure(A))                     # A > B #
   Direction:
        eq(circ_pressure(A),-)                 # circ_pressure(A) decreasing #
        eq(circ_pressure(B),+)
```

Table 5. Rule indexed under predicate *circ_pressure*.

Some other rules are listed in Table 6. The first relates the presence of windows or openings along a straight line between two spaces to the possibility of visual access between the two spaces. The second relates the presence of an opaque object on a straight line between two spaces to creating a visual barrier between the two spaces.

R1
Direction:
eq(vis_pressure(B),+)
Rule:
Individuals:
[space(A)]
space(B)
space(C)
[opening(X_i) window(X_i)]
Topological Relations:
next_to(A,C)
in(C,Xi)
in(A,Xi)
straight_line(A,B,all(X_i)) # a straight line must run from A to B through all the X_i #
Conditions:
vis_pressure(A) > vis_pressure(B)
Direction:
eq(vis_pressure(B),+))
eq(vis_pressure(A),-))

R2
Direction:
eq(vis_pressure(B),-)
Rule:
Individuals:
object(X)
space(A)
space(B)
Topological Relations:
straight_line(A,B,X)
opaque(X)
Conditions:
eq(vis_pressure(A),+)
Direction
eq(vis_pressure(B),-)

Table 6. Rules indexed under predicate *vis_pressure*.

The rule application algorithm is specified in Table 7. Noteable features of this procedure are as follows.

1. A separate goal is set up for each valid instantiation of a *for all* clause. For example, the goal statement:
goal = visual_barrier(X,Y)

where entry(X) and for all Y where room(Y)
applied to the plan represented in Figure 1. would be expanded as follows:
goal = visual_barrier(entry,living)
goal = visual_barrier(entry,dining)
goal = visual_barrier(entry,kitchen) ...

2. Separate goals are temporarily set up for each valid instantiation of the remaining variables. The appropriate instantiation is determined by breadth first search - the first goal achieved is maintained and the rest discarded. For example, the goal statement:
goal = circ_reln(X,bed1,residents) where bathroom(X)
in the context of the plan represented in Figure 1 would be expanded to:
goal = circ_reln(bath1,bed1,residents)
goal = circ_reln(bath2,bed1,residents)

Application of the rules would determine that the goal *circ_reln(bath1,bed1,residents)* could be achieved in a single step, so this goal would be held and the goal *circ_reln(bath2,bed1,residents)* would be dropped.

3. Lists of each potential rule sequence associated with a goal are maintained. When the goal is finally achieved the list of rules that led to the achievement of the goal are stored with the goal. The remaining rule sequences are discarded. As we have already noted search is breadth first so the first rule sequence identified will be that which satisfies the goal in the least number of steps possible.

The rules actually convey very little information. It is the identification of the *individuals* involved and the relations that must exist between these individuals in order for the goal to be achieved that is important. The instantiation of each rule is stored with that rule. Once an appropriate rule sequence has been identified, information as to how the goal has been achieved may be displayed as in Figure 2.

4. Rules are indexed under both predicate and direction. Determination as to an appropriate direction is made by reference to the relation between the required value of a variable in the initial state and the current value of that variable. See Step D'4 of the Rule Application Algorithm.

5. Intermediate states are constructed on the basis of the current value of a variable and the direction of change of that variable associated with the rule applied. See Step 3 of BACK PROPAGATE.

6. It is necessary to constantly monitor the intermediate states being generated in order to ensure that the current rule sequence has not lead to a state that has been achieved by a more efficient means. This is accomplished in Step F1 of the Rule Application Algorithm. Rule sequences that lead to an already identified intermediate state are pruned from the search tree.

7. Topological relations are sorted in INSTANTIATE REMAINING (Step B3) so as to minimize the number of instantiations that need to be investigated

8. Some predicates may not be specified in the current representation of the plan specification. In this case the user is asked to specify such information (Steps B1 and C2).

9. A limit is placed on the number of rules that can be chained together in order to achieve a goal. This limit is user defined. See Step D'1 of the Rule Application Algorithm.

A1. FOR each *goal* on *goal_queue*:

B1. Expand any *for all* clauses. Set up a separate goal for each instantiation justified in the current context. Ask for any predicates not already available.

B2. FOR each of the expanded goals:

C1. Set *current_goal* to current goal.

C2. Expand any *where* clauses of *current_goal*. Set up temporary goal for each instantiation justified in the current context. Ask for any predicates not available.

C3. Set up *rule_sequence_queue*:

ie. FOR each of the expanded goals:

D1. Convert goal into an initial and final state specification.

D2. Associate an empty rule sequence elt with the goal. Set state of the rule sequence element to the final state obtained in D2.

D3. Add to rule sequence queue.

C4. Set *current_sequence* to first element in rule sequence queue. WHILE there are still elements on rule sequence queue:

D'1. IF number of rules in sequence = UPPER LIMIT

THEN

Set *required_sequence* of *current_goal* to nil and GO TO B2.

D'2. Set *current_state* to the intermediate state associated with the last element on the rule sequence.

D'3. Find a predicate in *current_state* that has a value that differs from its value in the initial state. Note that predicates not specified in the initial state are assumed to be set to 0.

D'4. Determine the direction that the unsatisfied predicate has to change in. Values are given in the table below:

Value of variable in current state	Value of variable in initial state	Required Direction
+	0	+
-	0	-

D'5. Retrieve rules of appropriate predicate name and direction from rule table.

D'6. FOR each rule retrieved:

E1. Establish appropriate binding of constrained individual. Initialize list of *current_bindings* to this. Set sequence_elt to CHECK_BINDINGS(*current_bindings, rule, sequence_elt*).

If CHECK_BINDINGS fails GO TO D'6.

E2. Set up list of remaining bindings.

ie.. Set *current_bindings* to INSTANTIATE_REMAINING(*current_bindings*, *rule*)

E3. FOR each element on current_bindings list:

F1. IF initial state is satisfied

THEN Set *required_sequence* of *current_goal* to current sequence and GO TO B2.

ELSE

IF intermediate state achieved through instantiation with current binding is not equal to any intermediate state already achieved in pursuing current_goal
THEN copy the current sequence, add *sequence_elt* and append to the end of the rule sequence queue.

D'7. Set *current_sequence* to next element on rule sequence queue.

CHECK BINDINGS (*bindings, rule, sequence_elt*)

1. Check that bindings are of appropriate type as specified in the list of individuals associated with the rule.
2. Check any topology relations involving just these variables and variables for which values have already been established.
3. BACK PROPAGATE(*binding, sequence_elt, rule*) for each element in bindings list. Check any environment conditions involving just these variable and variables for which values have already been established.
4. If the bound variable is a global variable add to list of global bindings associated with current sequence.

BACK PROPAGATE *(binding, sequence_elt, rule)*

1. Look up value of variable in current state.
2. Look up direction of variable in rule.
3. Back propagate the value of the variable as follows:

Current Value	Direction	Initial Value
+	+	0
0	-	+
-	-	0

If there is no constraint on the direction of the variable then the value of that variable will remain constant.

4. Maintain record of state achieved as part of the intermediate state specification to be associated with this application of the rule. Associate with *sequence_elt.state*.

INSTANTIATE REMAINING (*rule, sequence_elt)*

A1. Set *current_bindings* to *sequence_elt.bindings*. FOR each element on *current_bindings*:

B1. IF there are no remaining uninstantiated variables
THEN RETURN(*current_bindings*)

B2. Set *future_sequences* to nil.

B3. Sort topological relations in order of the number of uninstantiated variables (least to most).

B4. Set *relation* to the topological relation with the minimum number of uninstantiated variables and (secondarily) maximum number of total variables.

B5. Find any sets of values that will satisfy the uninstantiated variables in that relation.

B6. FOR each set of variables found:
IF variables pass CHECK BINDINGS(*bindings,rule, sequence_elt*)

THEN
C1.Set *temp_elt* to CHECK BINDINGS and append current set of variables to *temp_elt.bindings*.
C2.Set *temp _sequences* to INSTANTIATE REMAINING (*rule, temp_elt*)
C3. IF *temp_sequences*, THEN append to *future_sequences*.
A2. RETURN (*future_sequences*).

Table 7. Rule application algorithm.

Sample Application

In previous sections I have introduced goal packages, artefact descriptions and a mechanism for mapping between the two. In this section I wish to consider the nature of this mapping process in the context of criticizing an architectural plan. The procedure is essentially as follows: an artefact description is presented for analysis, the goal packages relevant to that artefact are retrieved, and an attempt is made to establish a link between aspects of the artefact

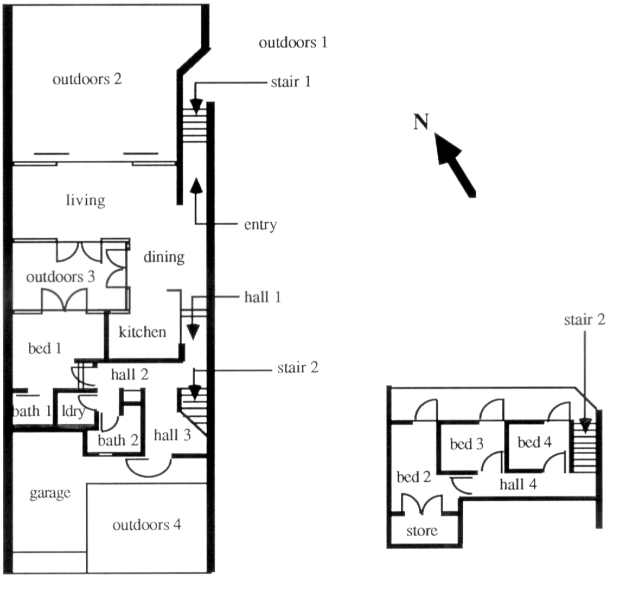

Figure 1. Sample floor plan.

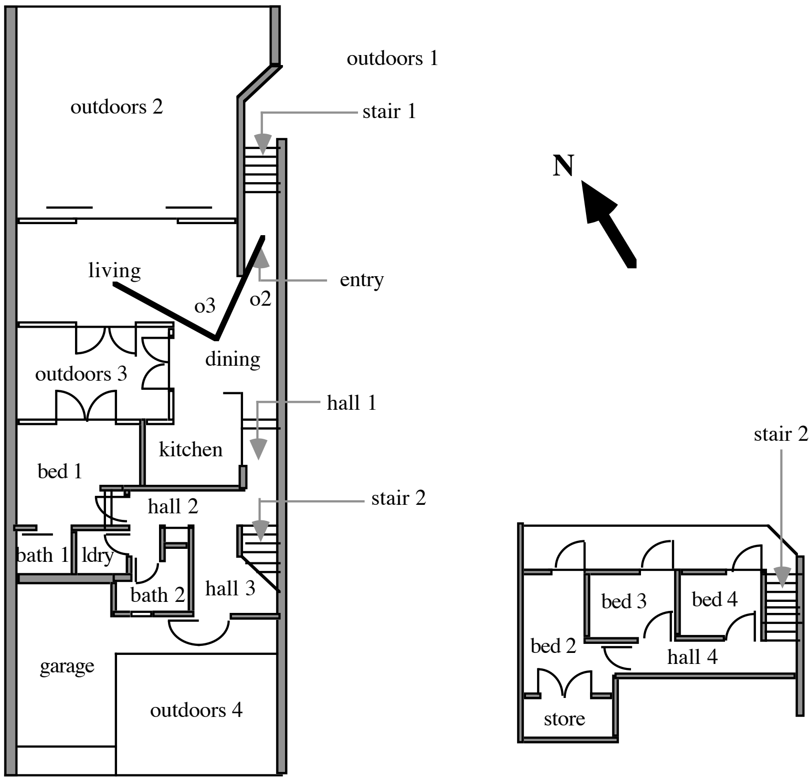

Figure 2 Diagrammatic representation of the features of the plan illustrated in Figure 1 relevant to achieving the goal *circ_reln(X,Y) where living(X) and entry(Y)*. The dark lines connecting entry and dining, and living and dining indicate *next_to* relations between these rooms. The labels *o2* and *o3* are the identifiers associated with the openings that allow passage between each of these pairs of rooms.

and each of the goals in the relevant goal packages. The outcome of this process is a representation of the goals achieved together with an indication as to those aspects of the artefact relevant to the achievement of these goals, and a list of the goals not achieved together with design recommendations pertaining to those goals.

Consider a situation where the plan illustrated in Figure 1 and the following contextual information is presented:

context = inner_city, medium_density
building type = residence
climate = temperate
goal = maximize_visual_privacy
goal = optimize_thermal_environment
occupants = family

The goal packages indexed under the indices *building type = residence, occupants = family* (Table 1) and *goal = optimize_thermal_environment, climate = temperate* (Table 2) would be retrieved. Application of inference rules such as those described in the previous section would yield the fact that a number of the goals contained within the goal packages retrieved are satisfied in the current plan. For example the goal *circ_reln(X,Y) where living(X) and entry(X)* contained within the goal package *building type = residence, occupants = family* is determined to be achieved by a) identifying the fact that there is a room labelled living that satisfies *living (X)* and a room entry that satisfies *entry(X)*; b) converting the goal *circ_reln(living,entry)* into an initial state *eq(circ_pressure(living),+)*, and final state *eq(circ_pressure(entry),+)*; and c) applying the rule indexed under the predicate *circ_pressure* (Table 5) twice, once to obtain an increase in the *circ_pressure* associated with the room *dining* from 0 to +, and a second to obtain the required increase in the *circ_pressure* associated with the room *entry* from 0 to +. The individuals and topological relations determined to be relevant to the achievement of the goal are listed in Table 8. A diagramatic representation of the features and relations that are relevant is illustrated in Figure 2. This diagram is intended to summarize the fact that it is the adjacency of the rooms entry and dining, and dining and living, and the existence of a means of passing between each pair of rooms that allows the goal *circ_reln(X,Y) where living(X) and entry(X)* to be achieved. A more comprehensive representation of the features of the plan relevant to the achievement of the goals contained in the two goal packages is illustrated in Figure 3. For simplicity's sake only the constraints imposed by the goal types *circ_reln*, *visual_access* and *sunny(_,winter)* have been displayed.

room(entry)
room(dining)
room(living)
opening(o2)
opening(o3)
next_to(entry,dining)
next_to(dining,living)
in(entry,o2)
in(dining,o2)
in(dining,o3)
in(living,o3)

Table 8. Symbolic representation of the features of the plan illustrated in Figure 1, relevant to achieving the goal circ_reln(X,Y) where living(X) and entry(Y).

It is envisaged that the principle advantage of such a display is that it would provide a compact representation of those aspects of the plan that are desirable and that the designer may wish to attempt to preserve if modifying the original plan. The utility of such a display would undoubtedly rest on the provision of facilities for grouping and ungrouping the constraints that arise from each goal. For example, the diagram in Figure 3 represents only a subset of the constraints identified in the context of the two goal packages retrieved, those displayed have been selected by goal type. Furthermore the overlay of information in Figure 3 obscures the association of particular features with a particular goal, much more clearly illustrated in Figure 2.

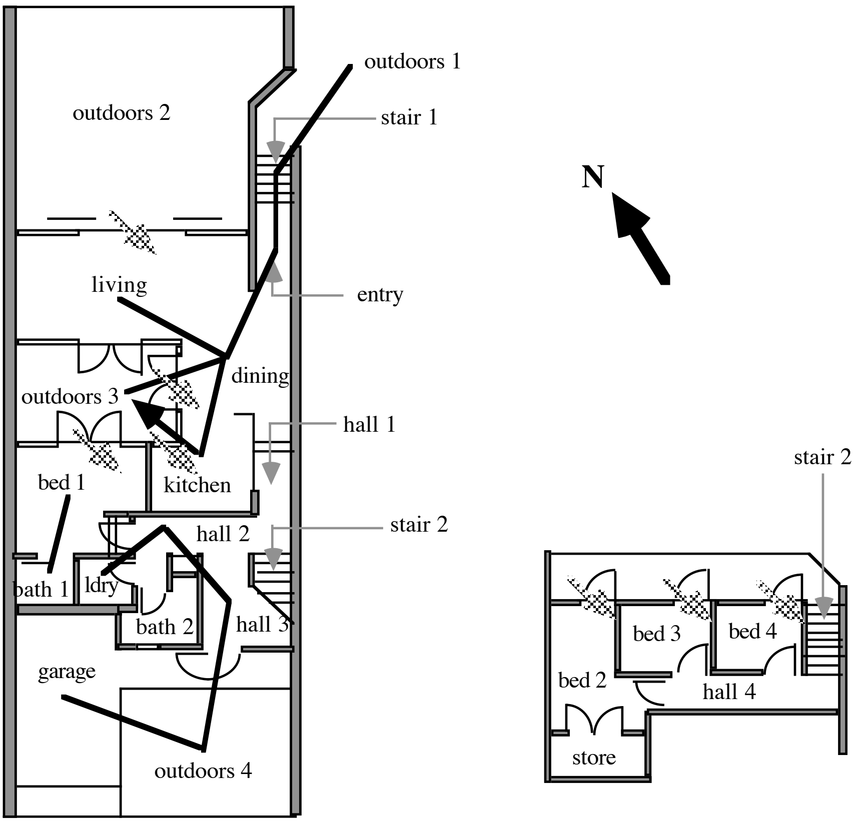

Figure 3 Diagrammatic representation of constraints gleaned from goal packages building type = dwelling, occupants = family and goal = optimize_thermal_environment, climate = temperate. Only the constraints imposed by circ_reln, visual_access and sunny(_,winter) are shown in order to keep the diagram simple. As in Figure 2, heavy lines between two spaces indicate a next_to relation. The dark arrow between the kitchen and outdoors 3 indicates the straight line access through the kitchen window required to create a visual relation between the two spaces. The grey arrows indicate windows or openings that permit the admission of winter sun to rooms where winter sun is required.

Perhaps more useful than information relating to the achievement of particular goals is a determination as to the goals that have not been achieved. Such a determination allows the relevant design recommendations to be presented to the designer for consideration. Returning to the plan illustrated in Figure 1, the initial classification of rooms is such that the goal *room = family(X)* is not satisfied. The following recommendations associated with this goal would be retrieved:

Residents need a separate living area to accommodate informal family activities -T.V., games, children's play, teenager's entertaining, and so on - which otherwise impose on more formal rooms. Research indicates a need for two separate living areas - an informal "sitting room" den and a formal "parlor" living room...In smaller units, increasing the size of the eating area within the kitchen may achieve the same end...Dimensions of the user need for *informal living areas include the fact that the equipment for some of these activities needs to be left in place over time, and other activities want to take place simultaneously with those going on in more formal living areas.*

From here the designer has the option of ignoring the recommendation, modifying the plan so as to include a family room or nominating an existing room, for example, the dining room, as a dual purpose room, intended to house the activities associated with both dining and family areas. If this latter course of action is taken other goals that refer to the family room will have to be evaluated in the light of this new information.

A second example of a goal determined to be violated is the goal *goal = circ_reln(X,Y, (residents,formal_visitors,informal_visitors)) where bathroom(X) and for all Y where [living(Y) or dining(Y)]* in the case where Y = living. An upper limit of three is set to restrict the number of rules that are chained together in attempting to achieve a goal. The goals:

circ_reln(bath1,living, (residents,formal_visitors,informal_visitors))

circ_reln(bath2,living, (residents,formal_visitors,informal_visitors))

are determined to be violated on the basis that a rule sequence of three or fewer members could not be found that reduced the difference between the states indicated by either of these goals. Violation of these goals leads to the retrieval of the following recommendation:

The bathroom must service the frontstage and backstage without privacy invasion of either stage...a visitor must be able to get from the sala to the bathroom without passing through the family room, kitchen, or bedrooms or any other private area.

The recommendation refers primarily to privacy requirements and the designer may decide that the relationship between *living* and *bath 2* is satisfactory. Alternatively an attempt may be made to create a more convenient relationship between the two types of spaces.

Conclusion

Much contention continues to surround claims as to the utility of computational tools in the development of architectural design proposals. This is due, at least in part, to the recondite nature of the processes employed by human designers in design development. In this paper attention has been focussed upon a circumscribed aspect of the expertise of the architectural designer, the evaluation of a proposal, and has where possible drawn on information explicitly available to assist in this task, namely design recommendation literature and plan drawings of existing artefacts.

Whilst the focus of this work was established by reference to a particular aspect of the expertise of the human designer, the simple symbolic model developed is not intended to simulate human capabilities in this area. The means by which a human designer determines the adjacency of two rooms, for example, is clearly much more sophisticated than the method described here. The goals considered are pragmatic, they are not sufficient to define the objectives of an architectural designer, but they do represent considerations that should not be ignored. Finally, no attempt has been made to evaluate the appropriateness of tradeoffs between goals - the objective has simply been to expose such tradeoffs.

In one sense the model proposed may be viewed as an attempt to articulate the role of rule-like expressions in the task of evaluating architectural floor plans. From this perspective several points are worthy of note. First, the application of the rules is context dependent, and an appreciation of context is integrally linked to the designer's ability to construct rich personal simulations of artefact and context. An attempt has been made to respect this fundamentally human capacity by simply exposing value judgements rather than attempting to automate the execution of these value judgements. Second, the rules employed by individual designers may be idiosyncratic. Emphasis must be placed on allowing a designer to construct personal goal packages. Third, the rules are not prescriptive, the implications of the rules can only be assimilated in the light of their application to particular artefacts in particular contexts. Fourth, the information that may be gleaned from application of the rules is modest in comparison to yielded by simulation of the proposal. On a more practical level, it is anticipated that the model may provide a basis for the following types of application:

1) A tool for student designer's to explore the tradeoffs made in existing plans, to develop a feel for the appropriateness of the application of various rules within specific contexts.
2) A critic integrated into a design environment. As outlined in the sample application, the model provides a means of conveniently applying and retrieving qualitative design recommendations as well as highlighting desirable aspects of a proposal and indicating relations that a designer should attempt to preserve. Such information may be of use to the novice designer who is still in the process of integrating information such as that contained in the goal packages into a perspective from which they can evaluate developing proposals.

Finally a point should be made about the role of evaluation in architectural design activity in general. It is the view of the author that evaluation is not simply an activity that occurs after a candidate has been proposed, but that evaluation of existing artefacts is fundamental to the proposition of a candidate artefact. Future work will involve using the methods described here to identify those aspects of a plan relevant to the achievement of particular goals, then using similarity based generalization methods in an attempt to identify consistencies across the forms of the resolution of these goals in particular contexts. Such an approach may be seen as an attempt to exploit the value judgements inherent in existing artefacts in order to isolate material that will provide a basis for an initial proposal appropriate to a given context.

References

Darke, J. 1979. 'The primary generator and the design process." *Design Studies* 1, no. 1: 36-44

Forbus, K.D. 1984. "Qualitative process theory." *Artificial Intelligence* 24 : 85-168

Foz, A. 1973. "Observations on designer behaviour in the parti." *Design Research and Methods* 7 no. 4: 320-323

Hillier, W., Musgrove, J. and O'Sullivan, P. 1972. "Knowledge and design." In Mitchell,W.J. (ed.), *Environmental Design: Research and Practice*, University of California.

Zeisel, J. and Welch, P. 1981. *Housing Designed for Families: A Summary of Research*. Cambridge, MA: Building Diagnostics.

NSB 21 (1971) *House design for temperate climates*. Australian Government Publishing Service.