DEDUCTION BASED REGION-USE PLANNING

Robin B. Stanton
Computer Science Department
Australian National University

GPO Box 4
Canberra, ACT Australia 2601
rbs@anucsd.oz@seismo.css.gov

Hugh G. Mackenzie
Division of Information Technology
CSIRO

GPO Box 664
Canberra, ACT Australia 2601
hgm@anucsd.oz@seismo.css.gov

ABSTRACT

This paper describes a deductive approach to "region use" planning problems. The planning task is viewed as one of devising a complete set of policy rules by iteratively modifying the rules and making their deductive consequences readily available to the planner. The application of this approach to a National Park region zoning problem is also described. Central to the system are the representation of management policy as knowledge based rules and the detection of inconsistencies in the rules. The implementation of the system has three closely coupled components; a spatial database, an interactive graphics server and the deduction subsystem. The last of these is briefly described in the paper.

1. INTRODUCTION

This paper describes a planning system which guides the evolution of a consistent set of management policy rules for a collection of regions. In addition the system deduces zoning maps for the regions from the policy rules, thereby guaranteeing that zonings satisfy policies. Planning tasks of this kind are common in land use management where the spatial extent of facilities and the separation of incompatible activities are important elements in controlling land use.

In its simplest form, region use planning involves the partitioning of an area (usually a geographic area) into zoned regions, where a zone stands for a set of permissions controlling activities in the associated region. The planning problem then can be construed as a search for single valued assignments of zones to regions which do not violate policy criteria.

In practice, the construction of zoning maps involve planners in a range of unformalised practical considerations, political intrusions and conflicting demands, all of which have to be allowed for in the formulation of policy. Under these circumstances inconsistent policies can arise easily and are often difficult to detect. The system described below can carry out the central task of ensuring that zone maps are consistent with policy rules. The resolution of inconsistencies is carried out by planners working interactively with the system. The aim is to construct a planning assistant rather than an expert planning system per se.

The principal theme is that as planners develop and refine policy rules, they should have ready access to the deductive consequences of the rules viewed as constructors of, or constraints on, possible zoning maps. This theme is allied with the aims of expert systems and includes an emphasis on explanations of consequences of the policy rule base.

The inferencing component of the planning system is called SPECS. SPECS assumes that planning tasks proceed from a collection of attributes of an area to be zoned and a collection of policy rules which a planner is required to follow in arriving at an overall zoning map. SPECS is one of three components in the planning system; the other two provide database and interactive graphics support.

The planning system is a general tool, and is being applied to problems in National Park planning and Urban planning. The following section describes Reefplan, a particular application which is used throughout as an illustrative example.

2. THE REEFPLAN APPLICATION

The Great Barrier Reef Marine National Park consists of five sections, one of which is called the Cairns Section. The problem of planning the Cairns Section has been studied by Cocks et.al.(Cocks 1982), using an approach based on LUPLAN (ibid), a "land use" planning system developed at the CSIRO. The Cairns Section is covered by approximately 500 regions which have been designated "atomic" regions for the purposes of assigning zonings. Typically each region contains a geographic feature such as a reef or shoal, an expanse of water closely associated with such features, or an area identified for its existing uses (such as shipping, fishing, etc.). The base data consists of approximately 17000 attributes and the boundary coordinates for the 500 regions. Figure 1 shows the regions in the Cairns Section, the westerly border being the coastline and the easterly boundary the continental shelf.

Activities to be controlled by zoning include fishing, swimming, coral and shell collecting, trolling, sight-seeing, recreational boating, and so on. The relationship between zones and activity permissions can be found in Cocks 1982. More or less in order of increasing permissibility, the zones are named Preservation (P), Scientific Research (SR), Marine National Park Type A and B (MNPA and MNPB), General Use Type A and B (GUA and GUB). (These abbreviations are freely used below).

Policy rules are illustrated by the following examples taken from Cocks 1982.

> A.1. All mapping units [regions] classified as SC (coastal shipping lane) or ST (transverse shipping lane) will be zoned General Use (Type A) (GUA).

> B.7. As far as possible, ensure that areas associated with existing off-shore national parks or within 5 km of existing coastal national parks and nature reserves are zoned as MNPA, MNPB or S.

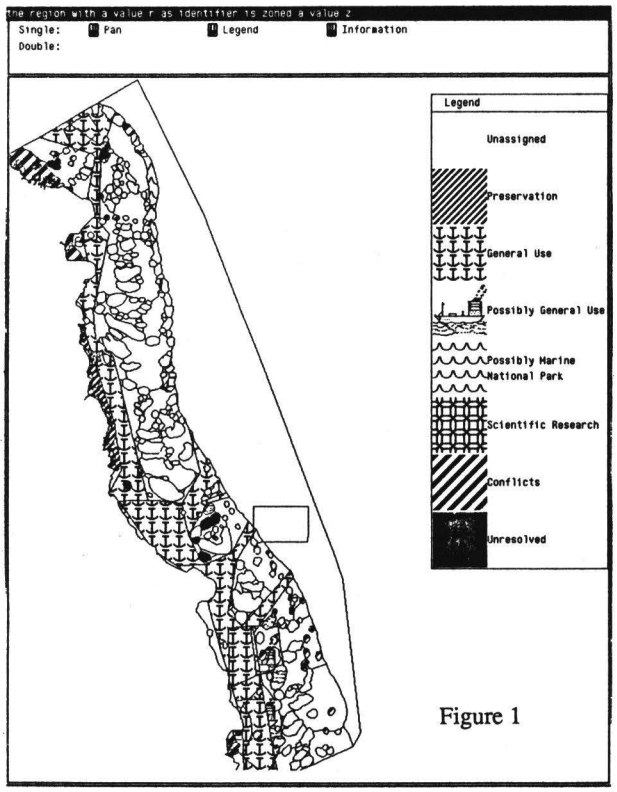

Figure 1

B.11. As far as possible, ensure that inter-reef areas are not zoned GUA.

B.12. As far as possible, ensure that fringing reefs and near shore areas are zoned MNPA or MNPB.

There are some 50 rules of this kind represented in the current system. In the context of standard AI techniques the following points about the rule set are important.

a. Many rules have disjunctive consequences
b. Some rules have negated consequences
c. Many rules contain spatial predicates over regions
d. There is no precedence ordering on the rules
e. Many rules contain the qualifier "as far as possible"

The consequences of a and b for the SPECS inferencing procedures are discussed in section 3.

The need to compute a range of spatial relationships over relatively large amounts of coordinate data (point c) led to the construction of a spatial database system using techniques described in Abel 1984.

Taken together, points d and e finesse the problem of consistency in policy making since any "as far as possible" (AFAP) rule can be rejected if otherwise it would lead to conflict. In fact the overall approach to mechanising a planner's assistant is mainly determined by how these AFAP rules are handled. Our initial approach was to regard such rules as assumptions, instances of which could be retracted to avoid inconsistencies (in the style of a truth maintenance system). This approach captures the permissiveness of AFAP policies in a fairly direct way but does not serve the goal of assisting a planner to develop definite policies which in themselves determine a zoning map.

In the LUPLAN scheme, policy rules are translated into components of an objective function which form the basis of a hill climbing search for a zone map of "maximum goodness". Goodness here is a measure of the extent to which regions which are suitable for certain activities are zoned so as to permit those activities to take place, and vice versa for activities which are not suited to particular regions. The search begins from a user supplied "initial zone map" which provides a point in the search space where the climb begins. Different initial maps lead to different final zonings, as do variations in the coefficients provided for the objective functions.

The SPECS solution to the zoning problem is quite different from the one used by LUPLAN. For example, in both systems region data taken together with given policy rules may result in many regions not being assigned a distinct zone. However the LUPLAN approach accepts this low level of determinism and proceeds to find local maxima on a surface defined by coefficients relating rules and region attributes. In contrast, the SPECS approach is to assist the planner to complete the rule set so that the zone to be assigned to each of the regions is determined by the rules, strengthening them where necessary to avoid inconsistencies. Thus the approach is to help planners refine and develop policy to the point where the problem is determined by the rules rather than offering one of many solutions found in a highly underdetermined space.

The extent to which most of the indefiniteness can be removed from typical region use planning tasks by this method is an experimental matter. Planners use intuitions and make aesthetic judgements in arriving at final zoning maps and the extent to which such knowledge can be formalised is unclear. However, by seeing the planner's task as one of decision making through rule strengthening, the SPECS approach provides a base for exploring formal justifications for giving regions particular zones.

3. Rule and Answer Representation

SPECS uses a set of backward chaining rules to compute relations, in a manner similar to Prolog, except that relations are computed in quasi-parallel, using standard techniques similar to those described in Charniak 1980. Operators in rule antecedents include the usual Prolog operators, together with the logical extensions described below, and general procedural escapes which can be used to access SQL, the graphical interface, and the spatial database.

A relation computed by SPECS is a set of 'tuple' objects, each of which has the following form, where the θ and φ are substitutions.

* θ, a simple tuple generator derived from the standard type of rules which do not contain constructive negations or disjunctive consequents.

* <θ **except** φ_1, φ_2, ...>, an exception tuple generator derived when a constructive negation operator appears in the antecedent of a rule.

* <**choose** θ_1, θ_2 ...>, a disjunction tuple generator, from rules with disjunctive consequents.

Given a set of variables $\{x_1, x_2 ...\}$ a simple tuple generator represents the set of ground instances of the term "**tuple**(x_1,

x_2, ...)" under the substitution θ. SPECS is explicitly designed for grounded answer relations so questions of interpretations of variables will not be discussed here.
The usefulness of constructive negation arises where is not convenient or efficient to fix the domain of interpretation for an expression. For example in "**not** P(x) ^ Q(x) .." we wish to restrict the tuples satisfying Q(x) to those not satisfying P(x) without enumerating the domain of P.

Rules B.7 and B.12 in Section 2 have disjunctive consequents. Such rules are not represented in that way but as standard rules which have a consequent which is the "most specific" generalisation of the set of disjunctions, and with an 'exclusive or' expression in the antecedent. This syntactic constraint was imposed for efficiency in searching the database. The xor operator constructs "multiple worlds", one for each of the disjuncts. Subsequent unifications take place independently within each of the worlds. Thus rule B.12, represented by:

fringereef(_r) ^ (_z = MNPA XOR _z = MNPB)
=> zone(_r,_z)

can produce a disjunction tuple generator in which two new worlds are established, the associated substitution for one containing MNPA for _z and the other, MNPB. Disjunction tuple generators also carry identification of the choice points which produced them, allowing split worlds to be reunited by an answer processing operation which carries out case analysis. In this way SPECS can deduce "zone(r1,P)" from the three rules,

dugong_habitat(_r, high) => zone(_r, P)
dugong_habitat(_r, medium) => zone(_r, P)
=> dugong_habitat(_r, medium) or dugong_habitat(_r, high)

since both worlds generated by the choice point in the third rule eventually contain "zone(r1,P)".

Rule B.11 in Section 2 has a negated consequent. Rules of this form are supported in SPECS by constraint rules of an "if deduced" kind, where demons prohibit the existence of certain answer tuples.

Thus we could represent rule B.11, (naively "inter-reef(_r) => not zone(_r,GUA)") as a demon which checks for the generation of an assertion unifying with "zone(_r,GUA)" and allows it if it is not the case that "inter-reef(_r)" for that unification.

We have also investigated using constraint rules to represent disjunctive consequents. A rule with the disjunctive consequent 'zone(_r, P) or zone(_r, GUA)' could be formulated as such a constraint rule, which would prohibit a tuple to enter the relation unless it unified with 'zone(_x,P)' or 'zone(_x, GUA)'. However, many of the rules with disjunctive consequents are intended to be constructive in the sense of allowing a planner either to choose freely between alternatives on some external criteria, or to remove disjuncts where they can be shown to lead to conflicts in a zone map (cf. assumption based reasoning, Doyle 1979, DeKleer 1986, Stanton 1985). In either case the constraints view of disjuncts is inappropriate. On the other hand, a classically complete treatment of disjuncts was ruled out, as it usually is, by efficiency considerations. The solution adopted in the current version of SPECS is to carry elements of a disjunct as if they were full members of the associated relation except that they are marked as "maybe" elements in the form {**choose** θ_1, θ_2 ...}. Before otherwise surfacing in an answer, such elements may have been deleted due to the existence of an unmarked equivalent element thus capturing one aspect of the intended meaning of rules such as B.12. The problem of the removal of alternatives which lead to conflicts is discussed in the next section.

4. Planning Cycle

A planning cycle consists of computing the "zone" relation (a binary relation relating regions, represented by small integers, to zones, represented by P, GUA, etc). To produce a zoning map, the relation must be a function. During the development of a plan however, the relation might not not be a function because of rules assigning multiple zonings to some regions and no zonings at all to others (see Davis 1985). Incremental removal of these inconsistencies is the basis of the planning cycle.

After computing the zone relation, its derivation may be examined textually or graphically (see Figure 2). Sets of zone values are associated with colour, or "textures" in the case of monochrome screens. Ambiguities are shown as described below. For any region, the rules involved in the zoning for that region, the zone values derived, and the base data held for the region may be examined by graphically selecting the region.

The derivation tree for any tuple generator may also be examined by means of an explanation sub-system. The explanation system also explains why tuples were not derived. Following the use of these tools to examine the relation, the underlying rules or base data may be modified to remove conflicts, and the cycle repeated.

In order to display a relation it is regarded as a function and the dependent and independent variables identified. In Reefplan, "zone(_r,_z)" is consideded to define z as a function of r. Then equivalence classes of coreferent tuple generators are found. Tuple generators are coreferent if the independent variables unify (in Reefplan, the variables must refer to the same region). To make this possible for the disjunction and exception tuple generators, we require that all substitutions in these tuples have the same independent variable values. Two tuple generators are in conflict if some grounded variant of each one violates the functionality condition. That is, the independant variable terms from both tuples unify with MGU θ, and the dependent variable terms unify when θ is applied to them. These classes are examined to detect conflicts and to remove choices which, a priori, cannot be made without producing a conflict. For example, if the class of coreferent tuples for _r =1 is:

{(choose <r=1, z=P>, <r=1, z=GUA>),
(choose <r=1, z=P>, <r=1, z=MNP>) }

the relation would be transformed to show <r=1, z=P> as the only possibility. In addition, if <r=1, z=SR> were in the class, there would be no consistent choice and the relation would be transformed to indicate a conflict. A general solution to the rejection of alternatives which lead to conflicts involves selections among the maximally consistent subsets of alternative disjunct tuples, a step we have resisted until more experience of the planning problem has been obtained.

Following this relation processing, regions whose independent variable values are in conflict have that conflict indicated by a previously unused colour or texture; those where a choice is still possible are indicated in the same way.

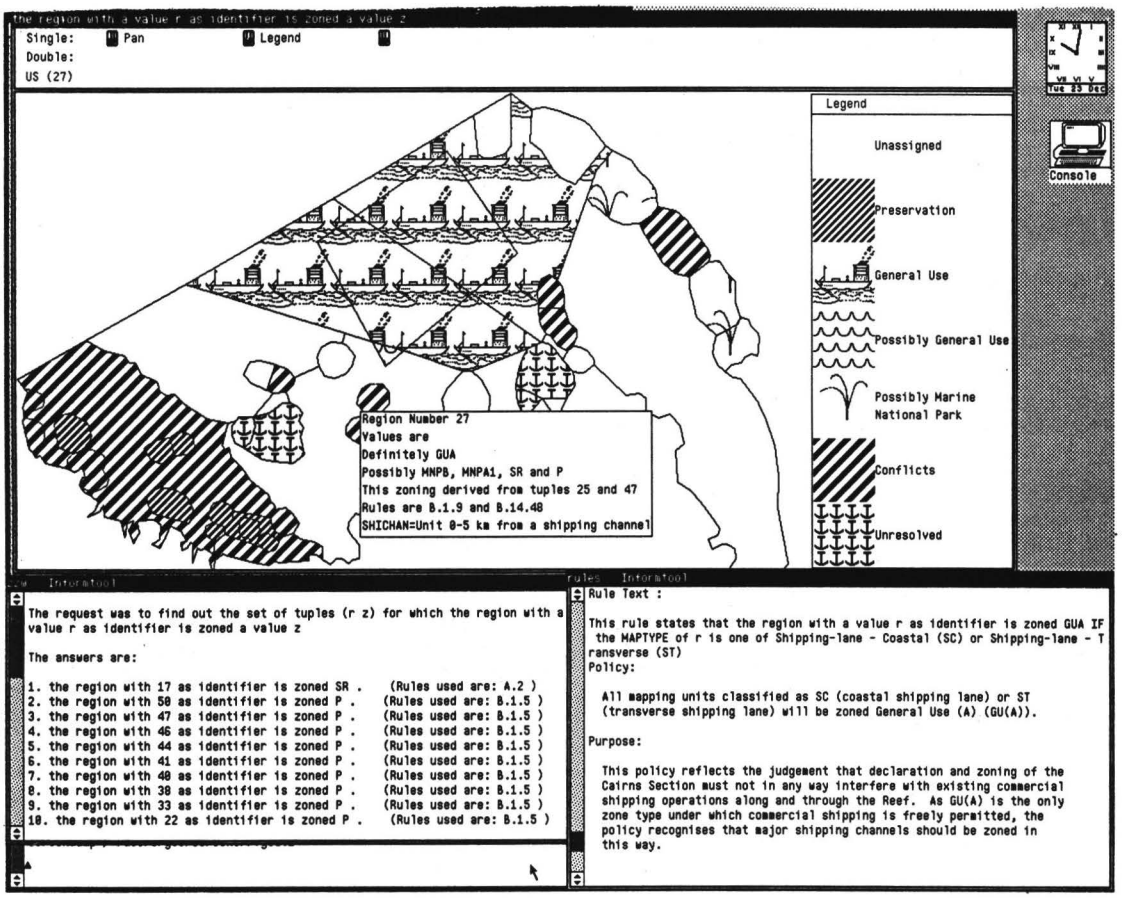

Figure 2

5. SUMMARY

The research reported here aims to prove the efficacy of a deductive approach to region use planning problems. It views the planning task as one of devising a complete set of policy rules by iteratively strengthening the set of rules and making their deductive consequences readily available to the planner. This thesis is being explored on four fronts: rule forms for expressing policy and associated reasoning strategies; detection of rule set completeness; efficient spatial inferencing techniques over large databases; and interactive graphics services tightly coupled to the deduction processes.

Experience with the SPECS system so far has centred on the Reefplan application. To obtain appropriate expressive power for this application, a form of disjunctive consequence and constructive negation were developed as enhancements to the familiar positive literal consequent rule structure. Although these enhancements are restricted to a ground case in the current version of SPECS, they proved to be adequate for the Reefplan problem.

6. ACKNOWLEDGEMENTS

The work reported in this paper is part of a joint project on "spatial inferencing systems" between the Computer Science Department, ANU and the Division of Information Technology, CSIRO, and is supported in part by an Information Technology Grant. John Smith, CSIRO, directed the construction of the spatial database interface. Colin Keith, Greg Toomey and Scott Milton made substantial contributions to the implemention of the system. Thanks are also due to Rob Edmondson for many discussions about non-floundering negations in logic programming.

REFERENCES

ABEL, D.J. and SMITH, J.L. (1984) : "A Data Structure and Query Algorithm for a Database of Areal Entities" *Australian Computer Journal* 16 4 (Nov), pp 147-154

CHARNIAK, E., RIESBECK, C.K. and McDERMOTT, D.V. (1980) : "Artificial Intelligence Programming" LEA, Hillsdale, NJ 1980

COCKS, K.D., BAIRD, I.A. and ANDERSON, J.R. (1982) : "Application of the SIRO-PLAN planning method to the Cairns Section of the Great Barrier Reef Marine Park, Australia", Technical Memorandum 82/1, CSIRO Division of Water and Land Resources, 1982.

DE KLEER, J. (1986) : "An Assumption Based TMS" *Artificial Intelligence* 28, pp127-162

DOYLE, J. (1979) : "A Truth Maintenance System" *Artificial Intelligence* 12 (1979), pp 231-272

STANTON, R.B. (1985) : "Propositional Rule Systems with Assumptions" *Australian Computer Science Communications* 7 1, pp15.1-15.13