The integration of expert knowledge in decision support systems for facility location planning

Citation for published version (APA):

DOI:
10.1016/0198-9715(95)00026-7

Document status and date:
Published: 01/01/1995

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 07. Oct. 2023
THE INTEGRATION OF EXPERT KNOWLEDGE IN DECISION SUPPORT SYSTEMS FOR FACILITY LOCATION PLANNING

Theo A. Arentze, Aloys W.J. Borgers, and Harry J.P. Timmermans

Urban Planning Group, Faculty of Architecture, Building and Planning, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

ABSTRACT. The integration of expert systems in DSS has led to a new generation of systems commonly referred to as knowledge-based or intelligent DSS. This paper investigates the use of expert system technology for the development of a knowledge-based DSS for the planning of retail and service facilities. The forms of knowledge involved in planning tasks are identified and organized in four layers. The layers describe the states and events in the facility system (the domain layer), their interrelationships (the inferential layer), procedures for solving well-defined subproblems (the task layer) and strategies for approaching the overall problem (the strategic layer). The potentials of decision tables for representing qualitative and complex knowledge is discussed and illustrated with applications in the field of retail planning. It is shown that expert knowledge from each of the four layers can be used to improve the modeling capabilities and intelligence of a DSS. The result is a powerful and flexible DSS that supports planning at the domain, inferential, task and strategic level dependent on the preference of the decision maker and characteristics of the problem.

INTRODUCTION

Existing expert system techniques provide the means to model domain-specific knowledge so that it can be used and maintained in a computer for decision making. The aim of this paper is to assess the possibilities of expert system techniques for spatial planning. Hereby, we focus on the problem of the planning of facilities that serve a spatially dispersed population, which is a concern of local governments and commercial firms who wish to develop or reorganize a facility network. The type of facility system we focus on is defined by Leonardi (1981) as users-attracting systems, where people determine the choice of facility destinations and travel to destinations (in contrast to delivery systems). Examples of users-attracting systems are some school systems, some health care systems, retail systems and recreational services. The planning tasks include elements of monitoring demands–supply relations in the system (monitoring task) and developing plans when disequilibria (diagnosis and repair task) or opportunities for expanding a service network (design task) arise.

Based on theoretical considerations and cumulated experience in expert system development a set of conditions has been established for determining whether a problem is appropriate for expert system solutions (Parsaye & Chignell, 1988, p.299; Luger & Stubblefield, 1989, p.296; Prerau, 1989; Ortolano & Perman, 1990). The most fundamental condi-
tion is that the problem domain is well-defined. Clearly, the facility planning problem does not meet this condition for two reasons. First, the knowledge domain comprises a number of different disciplines or dimensions. For urban and regional planning in general, Masri and Moore (1993) mention economic aspects, legal and constitutional issues, social demographics, moral guidelines and technical attributes. Because the knowledge rules regarding each of these dimensions are not clearly defined, the knowledge domain is not sufficiently self-contained and narrowly focused. Second, the location planning problem is typically ill-structured because decision makers do not have complete or reliable information regarding goal specifications, decision alternatives and their outcomes (Densham, 1991; Armstrong et al., 1991; Borgers & Timmermans, 1991; Hopkins, 1984). Particularly in governmental planning, there are often multiple possibly conflicting goals or interests groups involved so that compromise solutions must be found through weighting of goals or negotiation. Moreover, uncertainties often exist regarding the prediction of developments in the facility system and the generation of possible location strategies. Furthermore, it is often not possible to predict with certainty the impact of these autonomous or planner-defined changes on goal dimensions.

Given the lack of clear boundaries and structuredness of the knowledge domain, the development of an expert system (ES) for facility planning is not a feasible approach. Instead, it is better to focus on the development of a decision support system (DSS), which is explicitly designed to deal with unstructured or semi-structured problems. However, a DSS approach does not exclude the use of ES techniques. In many respects ES and DSS are complementary approaches. The recognition of this complementarity has led to the development of a new generation of systems that integrates ES and conventional DSS. These integrated systems have been proposed in operational research, management science, and spatial planning and have been referred to by different authors as intelligent DSS (IDSS), intelligent support systems, expert DSS (EDSS), expert support systems (ESS) and knowledge-based DSS (KBDSS) (El-Najdawi & Stylianou, 1993; Han & Kim, 1990a; Luconi et al., 1986).

There are several forms of integration that have been discussed in the literature (see El-Najdawi & Stylianou, 1993). For our purpose, we distinguish two major forms. First, ES techniques have been used for incorporating domain knowledge in DSS to enhance the modelling capabilities of these systems. The planning information system ISIS (Wright, 1990) illustrates this approach. ISIS combines ES, algorithmic and algebraic models for allocating land uses among specific land areas. ES models are used to assess the suitability of areas for locating an activity when dependencies (interactions) between suitability factors exist which can not be expressed in a mathematical model. Furthermore, several studies (Reitsma, 1990; Amer et al., 1993; Arentze et al., 1994a) have shown how ES can be used for site suitability analysis as part of a larger DSS for locating facilities.

Second, ES have been suggested for improving the intelligence of DSS, i.e. for: (i) creating an intelligent database that helps users in querying and searching the database; (ii) developing an intelligent interface to the model base of the DSS that helps users in selecting and interpreting the output of complex models; and (iii) building a friendlier user-interface for example by supporting communication through natural language (Ortolano & Perman, 1990; Han & Kim, 1990a). ES has proven to be particularly useful for interfacing the DSS model base (the second type of application). In the field of operational research, several systems have been developed that use ES for interfacing optimization models (e.g. linear programming) (Erickson et al., 1989; Elam & Konsynski, 1987; McGovern & Samson, 1989; Han et al., 1991). Armstrong et al. (1990) describe a knowledge-based DSS for locating facilities that provide comprehensive and far reaching support for using location-allocation models. The ES modules assist users up to the level of developing strategies for solving complex
The integration of expert knowledge in DSS for facility location planning

location problems. In interaction with the user the system develops a metaplan that determines the scenarios to be explored and models to be used dependent on the given problem. Also the intelligent planning system developed by Wyatt (1991) uses ES to support planning at the strategic level. These systems show the potential of ES for making the model base of the DSS accessible even for users who are not technical experts. In the knowledge-based DSS named XPlanner developed by Han et al. (1991), ES is used both for enhancing and interfacing the model base. XPlanner supports facility planning and management in military installations. The ES component performs classification and diagnosis tasks and serves to formulate and interpret a zero–one integer optimization model.

In this study we investigate the possibilities of using ES for improving the modelling capabilities and intelligence of conventional DSS for facility planning. To achieve this goal, the paper is structured as follows. First, we develop a model of the knowledge domain of planning tasks, to identify and classify the knowledge components involved. Next, in the third section, we discuss decision tables as a method of representing knowledge and discuss the role ES may play in modelling the different knowledge components identified. In the section that follows we focus on the integration of ES in a DSS for facility planning. Finally, we conclude with summarizing the conclusions and discussing ways for future research. Examples of facility planning problems used throughout the paper are drawn from the field of retail planning.

A KNOWLEDGE MODEL OF PLANNING

The planning of facilities is a complex problem which consists of several subproblems that differ in the nature of the knowledge involved. Therefore, it is probably not possible to solve the planning problem using a single modelling technique. Instead, a combination of algorithmic, algebraic and ES approaches is required. In this section we develop a model of the knowledge involved in planning, whereby a broad definition of knowledge is used to include data, heuristic knowledge and scientific methods. The model is based on a conceptual analysis of planning problems given by Arentze et al. (1994b) and the conceptual modelling language developed by Breuker & Wielinga (1989). Breuker & Wielinga distinguish four layers for describing expert knowledge in knowledge-based systems. From low to high level, these are the domain, inference, task and strategic layer. We will describe the knowledge and knowledge sources in facility planning for each of these levels in turn.

The domain layer

This layer, at the lowest level, contains static, declarative knowledge about the facility system being investigated. In our model, the planning problem is viewed as one of finding the set of actions that reduces discrepancies between an anticipated future state and a given goal state of the system. Therefore, a number of perspectives of the system are of interest in the planning task. Fig. 1 schematically shows the perspectives (boxes) and the way they are interconnected (arrows). Descriptions of this set of system perspectives are the elements of the domain layer. We now specify the contents of the perspectives.

The present state and the goal state of the system are given. Generally, the goal state represents the desired state of the system at some future moment in time. The anticipated future state of the system is the result of two types of anticipated changes of the present state in the plan period. First, there are expected developments in the system that are outside the control of the planner. These changes are referred to as autonomous developments, since they occur independent of planner’s actions. In the facility planning problem, we can
think of demographic, economic (e.g. real income) and technologic (e.g. new retail formulas) changes which are often beyond the control of the planner and yet relevant for the goals pursued. Second, there are anticipated changes that follow from action plans developed by the planner. In the facility problem, possible actions may include changes in service supply, such as the opening, closing, relocation or upgrading of facility centres. The discrepancy state is the result of the comparison between the anticipated future state and the given goal state; it shows in some way the mismatch between future state and goal state and, thus, indicates the extent to which the actions undertaken are successful in achieving the desired goal state (given the present state and the autonomous developments). Dependent on the specification of the goal state, discrepancies involve either imbalances between demand and supply or opportunities for expanding or otherwise improving a distribution network.

Usually, the present state and the goal state are described in terms of different sets of variables. The present state of facility systems is described in directly observable variables that are relevant for the goals pursued. In the following we will refer to these variables as $X$-variables. The choice of $X$-variables depends on the given goals and available data and model resources. In facility planning problems, the present state is usually described in terms of: (i) the location and attributes of residential zones of the consumer population (demand) and of facility centres (supply), (ii) the spatial structure (the road network) connecting locations, and optionally also (iii) observed interactions between demand and supply. On the other hand, the goal state is normally described in terms of decision-criterion variables. In facility planning problems, these $C$-variables (criterion variables) generally concern aspects of: (i) service coverage of demand, (ii) economic performance of supply, and possibly also (iii) mobility and traffic flows. Given the different sets of variables used, the anticipated future state (in $X$-variables) can not be compared directly to the goal state (in $C$-variables). Therefore, in an intermediate step the anticipated future state is evaluated in terms of $C$-variables, resulting in an extra system perspective, namely the future state in $C$.

Apparently, there are also different sets of spatial objects considered in the facility planning problem. Dependent on the goals formulated, the following groups may be relevant: (i) demand objects (residential zones), (ii) supply objects (facility centres), (iii) objects of the spatial structure (e.g. road segments), and (iv) interactions between demand and supply. Given these different groups there are different perspectives within each main perspective (present state, goal state etc.), such as for example the demand, supply and spatial structure of the present state. In Fig. 1 only the demand and supply side are depicted as possibly relevant subperspectives. It should be noted that not necessarily all system perspectives are relevant in specific planning problems. For example, if only economic goals are pursued, the demand side of the goal, discrepancy and future state in $C$ may be irrelevant. As another example, if the planner does not have the means to control the demand side of the system, the Action–Demand perspective has no meaning and is discarded.

Until so far it is assumed that the planner has the means to directly change the present state by acting on $X$-variables. For commercial planners (e.g. retail companies) this is normally the case regarding outlets of the own chain. However, governmental planners do not have the means to directly control private sector facilities. Instead, they are able to indirectly influence developments in these systems using financial incentives, zoning regulations or public sector investments. In this class of planning problems we need another set of variables to describe action plans, referred to as $I$-variables (instrument variables). Developing plans in $I$ is a complex task since there are usually more ways in which desired changes in $X$ can be accomplished by actions in $I$. At least in the Netherlands, governmental planners solve this problem in two stages. In the first stage, actions are planned in $X$ to determine the desired future state in $X$. Next, in the second stage, a final plan is developed in terms of
The integration of expert knowledge in DSS for facility location planning

present state in X

DEM SUP

future state in X

DEM SUP

action in X

DEM SUP

present state in C

DEM SUP

future state in C

DEM SUP

discrepancy state

DEM SUP

goal state in C

DEM SUP


I-variables whereby the desired future state in X serves as the goal state. So, in the second stage the plan developed in the first stage is made operational (in terms of implementable measures). This task is less complicated than the original task, since the goal state is described in the same set of variables as the present state (X-variables). The model of planning actions in I has a similar structure and is schematically shown in Fig. 2.

Having specified the relevant system perspectives, the knowledge elements at the domain level can now be expressed in predicate calculus (ProLog) notation, as follows:

\[
\text{domain - element(perspective}_i, \text{ objectlist}_i, \text{ variable}_{ij}, \text{ valuelist}_{ij})
\]

where: perspective\(_i\) is the \(i\)-th system perspective (e.g. the present state of the demand side); objectlist\(_i\) is the list of spatial objects considered in the \(i\)-th perspective; variable\(_{ij}\) is the \(j\)-th variable (e.g. a population attribute) of interest in the \(i\)-th perspective; valuelist\(_{ij}\) is the list...
Table 1. Domain Knowledge in Planning Tasks (Derived from Figures 1 and 2).

- autonomous developments
- present state of the system in X-variables
- goal state of the system in C-variables
- future state in X-variables
- future state in C-variables
- discrepancy state
- actions in X-variables
- actions in I-variables

Table 2. Inferential Knowledge in Planning Tasks (Derived from Figures 1 and 2).

<table>
<thead>
<tr>
<th>Function</th>
<th>Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>- inferring the future state in X from the present state, autonomous developments and actions in X or I</td>
<td>- causal knowledge: methods for forecasting and impact assessment</td>
</tr>
<tr>
<td>- inferring the future state in C from the future state in X</td>
<td>- conceptual knowledge: methods for system performance analysis</td>
</tr>
<tr>
<td>- inferring the discrepancy state from the goal state and future state in C</td>
<td>- conceptual knowledge: methods for system performance evaluation</td>
</tr>
<tr>
<td>- inferring actions in X or I from the discrepancy state in C</td>
<td>- instrumental knowledge: methods for plan generation</td>
</tr>
</tbody>
</table>

of values (or scores) of the objects on the ij-th variable. Associated with these knowledge elements are properties such as reliability, accuracy, and completeness. Table 1 summarizes the domain elements that describe the facility system. The elements are either given (e.g. present-state variables) or derived using inferential knowledge (e.g. future-state variables).

**The inference layer**

This layer, one level higher, contains procedural knowledge for making inferences. In our model, inference involves deriving domain knowledge elements from existing ones. Therefore, the inference layer specifies the dependency relationships that exist between domain elements. The knowledge elements at the inferential level are methods attached to dependent system variables for defining their values dependent on other variables, as follows:

\[
\text{inference} - \text{element}(\text{objectlist}_i, \text{variable}_{ij}, \text{method}_{ij}, \text{valuelist}_{ij})
\]  

(2)

where: method_{ij} is the ij-th method that defines the scores of the objects on the ij-th variable and other symbols are defined as above. Table 2 summarizes the knowledge elements that generally make up the inference structure in facility planning problems. Together, the set of methods defines the relationships between system perspectives as represented by the arrows in Fig. 1 and Fig. 2. Based on the type of function they perform, we can distinguish the following forms of inferential knowledge:

**Causal knowledge**

This type of knowledge describes the causal relationships between system events and states. In terms of the planning scheme, it performs the function of deriving the Future State in X based on the Present State in X, Autonomous Developments and Actions in X. Generic methods developed in the statistic and spatial sciences play a role in performing this function. Statistical methods, such as projection techniques, regression modelling and time series
analysis, can be used for forecasting demographic and socioeconomic developments. Spatial interaction or choice models are available for predicting interactions between demand and supply dependent on supply locations and attributes (under plan conditions). Furthermore, domain-specific knowledge, possibly heuristics, may be used for complimenting statistical forecasting methods or for forecasting developments of a qualitative nature.

**Conceptual knowledge**

This knowledge defines the conceptual relationships between different descriptions of the same system state. In the planning scheme, it performs the function of first deriving the Future State in $C$ from the Future State in $X$ and next determining the Discrepancy between this state and the Goal State. The first step involves determining system performance on social and economic dimensions ($C$-variables), given the predicted or planned location, attributes and interactions of demand and supply ($X$-variables). In the case of quantitative $C$-variables, numeric performance indicators developed in the spatial sciences may be used for this purpose. Otherwise, in the case of categorical $C$-variables, scientific or heuristic rules that assign (demand or supply) objects to performance classes are appropriate. The second step involves a comparison between observed system performance and objectives. Standard multicriteria evaluation methods of measuring deviations with an ideal point may play a role in this step. Alternatively or additionally, normative planning rules may be used that signal mismatches which ask for corrective actions.

**Instrumental knowledge**

Instrumental knowledge describes means-ends or teleologic relationships that relate operations (actions) to discrepancies between the system state and the goal state. In terms of the planning scheme, instrumental knowledge performs the function of generating optional Actions in $X$ or in $I$ that are likely to be successful for solving or reducing an observed Discrepancy (in $C$ or $X$, respectively). Instrumental knowledge describes possible location strategies ($X$-variables) or available instruments ($I$-variables) in terms of the cases they apply to, the conditions for their application and the effects they probably produce (as in the general problem solver program SOAR (Laird et al., 1987)).

The inference structure that is composed of these forms of knowledge describes what inferences can be made, but not when they should be made. The task layer one level higher describes how inferential knowledge must be applied for accomplishing planning tasks.

**The task layer**

In contrast to the inferential layer, which describes relationships between declarative elements, the task layer contains knowledge of a normative or prescriptive kind that specifies how a planning problem should be solved. A systematic way of solving planning problems consists of two stages. In the first stage, the problem is modelled by selecting the variables of interest and attached methods that describe the different system perspectives. Then, in the second stage, the model (inference structure) specified is used to develop an action plan first in terms of $X$-variables and, if the planner has no direct control on $X$-variables, next in terms of $I$-variables. Both stages are not trivial and require extensive higher-level, task performance knowledge. We distinguish analytic (for the first task) and task–procedural (for the second task) knowledge.
Table 3. Task Analytic Knowledge in Planning Tasks (Derived from Figures 1 and 2).

<table>
<thead>
<tr>
<th>function</th>
<th>typical issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>- selection of C-variables and conceptual methods dependent on the goal state</td>
<td>- which objectives operationalize goals and how should the objectives be measured?</td>
</tr>
<tr>
<td>- selection of X-variables and causal methods dependent on chosen C-variables and available data sources</td>
<td>- which factors determine system performance and how should the factors be predicted under future conditions?</td>
</tr>
<tr>
<td>- selection of dimensions of autonomous developments dependent on X-variables and available data</td>
<td>- which developments influence relevant factors and should therefore be monitored?</td>
</tr>
<tr>
<td>- selection of possible actions in X and I and instrumental methods dependent on chosen X-variables</td>
<td>- which actions or instruments are available to control relevant system factors and under what conditions do they apply?</td>
</tr>
</tbody>
</table>

Analytic knowledge

The variables and attached methods used for describing the different system perspectives should meet the information needs of the planner on the one hand and available data and method resources on the other. The problem of choosing variables and methods consists of a number of questions which are summarized in Table 3. The competence to solve these questions is referred to as analytic knowledge. This type of knowledge is described in methodological studies in the spatial and statistical sciences. These studies further describe properties of methods needed for interpreting analysis results, such as their reliability and accuracy.

Task procedural knowledge

The facility planning problem has a combinatory character and is typically ill-structured. Consequently, it is hardly ever possible to identify in a sure-fired way the set of actions that reduce (to an acceptable level) observed discrepancies. The instrumental knowledge available is at best sufficient for suggesting tentative actions that might be successful. Therefore, an action plan is typically developed through iteratively generating and evaluating action alternatives. Task procedural knowledge prescribes a sequence of inferential steps for achieving a goal (solving a problem). The elements of this layer are algorithms or programs for solving subproblems that are relatively well-structured. This conception of task-procedural knowledge corresponds to the subproblem-solver module in the KBDSS approach developed by Armstrong et al. (1990).

The development of task-procedural knowledge is the major focus of the field of operational research and AI. A number of problems that are encountered in spatial planning are addressed in these fields. Table 4 gives some examples. The algorithms or programs developed can be viewed as a layer that controls inferential processes one level lower. Consider for example the interchange algorithm for solving (a certain class of) location-allocation models. The algorithm starts with a random solution of facility locations and sequentially improves the solution by single relocations (substitutions) until the objective function has reached an (possibly local) optimum. In terms of the planning scheme, the cycles consists of generating Actions in X (single relocations), determining the Future State in X under plan conditions (location and allocation in the new solution), determining the Future State in C (the objective-function value) and evaluating the Discrepancy (compare the objective-function value with the current maximum) until the stop condition is met. After each it-
Table 4. Examples of Task Procedures in Spatial Planning Tasks.

<table>
<thead>
<tr>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>generating interesting action alternatives for further consideration</td>
</tr>
<tr>
<td>- identify candidate sites using potential measures</td>
</tr>
<tr>
<td>- generate location configurations using genetic algorithms</td>
</tr>
<tr>
<td>- generate a spatially heterogenous set of highly performing location</td>
</tr>
<tr>
<td>configurations using mathematical programming</td>
</tr>
<tr>
<td>reducing a given set of discrete action alternatives</td>
</tr>
<tr>
<td>- select the subset of feasible sites based on screening criteria</td>
</tr>
<tr>
<td>- select the subset of non-inferior sites using goal programming</td>
</tr>
<tr>
<td>- select a manageable number of alternatives based on optimization criteria</td>
</tr>
<tr>
<td>generating the optimum configuration of a facility network</td>
</tr>
<tr>
<td>- solve a standard location-allocation model using a heuristic algorithm</td>
</tr>
<tr>
<td>or mathematical programming</td>
</tr>
<tr>
<td>- determine the configuration that optimizes an objective function using</td>
</tr>
<tr>
<td>complete enumeration</td>
</tr>
<tr>
<td>ranking plan alternatives using standard</td>
</tr>
<tr>
<td>- multicriteria evaluation methods</td>
</tr>
<tr>
<td>- cost–benefit analysis techniques</td>
</tr>
<tr>
<td>- goal-achievement analysis techniques</td>
</tr>
</tbody>
</table>

eration (a relocation or so-called swap) the Present State is updated, to reflect the current solution, and the Goal State is set to the current maximum.

Since the objective function specified normally does not completely cover the set of planner’s objectives, the solution generated by an optimization algorithm or program is rarely taken as the final plan, but rather as an interesting plan alternative. In general, solutions at the task-procedural level are only steps in a larger problem solving process that is controlled by knowledge at a higher level.

**The strategic layer**

Complex (ill-structured) problems, like facility planning, require flexible problem solving adapted to information needs, styles and preferences of the decision maker and available data and method resources. At the strategic layer, a plan of how to solve the problem is developed dependent on these problem characteristics. The goal (solution of the problem) is subdivided into subgoals for which standard task procedures exist or that can be achieved by single inferential steps. Task or inferential elements are linked not in a static way but conditional upon results obtained. The progress of the problem solving process is monitored and alternative strategies are developed when procedural steps do not yield the desired results. So, the strategic layer dynamically determines the task structure at the lower level. Table 5 provides some examples of goals and corresponding strategies in retail planning. The strategies listed give a first decomposition in major steps. Every step should be further subdivided to the level where they match with goals at the task layer so that they can be carried out by standard procedures.

The strategic layer in this model corresponds to the metaplanner module in the KBDSS proposed by Armstrong et al. (1990). The metaplanner assists the decision maker in achieving a goal by suggesting steps that can be handled by the subproblem solver. Also in their system both the subproblem solver (in our model the task layer) and metaplanner (in our model the strategic layer) link procedures to goals. The difference is in the nature (algorithmic or strategic) of the procedures. As Armstrong et al. put it

"the subproblem-solver deals with a precisely stated analytic problem and uses an algorithm to solve the problem. On the other hand the metaplanner deals with an ill-specified goal where many important decision-making criteria may be implicit or even unknown at a
The evaluation of a proposal for locating a large-scale retail facility when a broad set of goal dimensions is involved (see Rogers, 1979)
1. determine short term and long term effects on goal dimensions (design and environment, traffic flow, consumer's interests, retailer's interests etc.)
2. determine corrective measures for alleviating negative effects (e.g. investments in the road network)
3. analyze the costs and benefits of the proposal
4. judge the desirability of the proposal

The development of a location strategy for expanding a retail chain when the objective is to optimize economic performance (see Ghosh & McLafferty, 1987)
1. select a market region (e.g. a city area) based on market expansion potential and degree of market saturation
2. select suitable market areas (e.g. a sector of a city) based on retail potential and level of competition
3. select candidate sites based on characteristics of the retail environment and site
4. generate and evaluate possible network configurations (number, location and size of outlets)
5. judge the desirability of the optimum configuration

The development of a long term policy for optimizing the service coverage and economic viability of supply of daily retail goods
1. predict demographic, economic and structural developments that affect demand-supply relationships in the daily good sector
2. analyze service coverage and economic viability of the daily good sector in the future market situation
3. generate actions (upgrading, closing, opening, expanding centres) to optimize service coverage under constraints of economic viability and preserving a well-formed retail structure
4. generate implementable measures (e.g. financial support, zoning regulations) to accomplish the optimum long term retail structure

In a DSS environment this difference is reflected in the type of user-system interaction. At the task level the system controls the problem solving process and may prompt the user for inputs, whereas at the strategic layer the system guides a user-defined process.

Applied studies that contribute to the systemization of spatial planning can be viewed as sources of strategic knowledge. Here, we will not attempt to summarize approaches that have been suggested, but instead we refer to the examples given in Table 5.

**Relations between knowledge layers**

The knowledge layers distinguished and the way they are interconnected are summarized in Fig. 3. The two lowest level layers describe the system to be planned in terms of system states or events (the domain level) and (causal, instrumental and conceptual) dependency relationships (the inferential level). The subsequent two layers describe how this knowledge
can be used to solve problems at the level of well-structured subproblems (task-knowledge) and the semi-structured overall problem (strategic knowledge).

It follows from the foregoing discussion that knowledge at a higher level relates knowledge elements at a lower level to each other. As Breuker and Wielinga point out, each layer constrains the processes that can take place at lower levels. The strategic layer constrains possible task procedures (e.g., sequences of tasks), which in turn determine possible inference processes and thus the possible system states and events at the domain level. Therefore, every layer acts as a further constraint to what can take place at the domain level. Since the domain layer includes the possible solutions (action events), this means that higher level knowledge restricts the search or solution space. The degree to which it does this depends particularly on the extent to which problems are structured at the task and strategic level. For example, if a non-intelligent strategy, such as "evaluate all possible actions and select the best one" is used, and the only task knowledge available states "if a variable is needed while it is unknown, then execute the attached method", virtually all domain-level processes that are allowed by the inference structure are possible. Adding intelligence to the strategic layer from this point will result in a meaningful reduction of the search space (and, consequently, in better outcomes).

From the point of view of designing DSS, the organization of knowledge in different layers is useful only if the layers distinguished can be specified to some degree independently of each other. Then, providing support to separated levels would strongly enhance the flexibility of the system. For example, if algorithms chosen at the task level leave open the choice of methods to be used at the inferential level, the system would be highly adaptable to preferences and styles of the decision maker and available data sources. To some degree this is probably the case in facility planning. For example, the choice of an interchange algorithm for optimizing a facility network allows various specifications of the allocation method (causal knowledge), objective function (conceptual knowledge) and the interchange heuristic (instrumental knowledge). Therefore, having selected the algorithm at the task level, the user is within certain limits free to specify components at the inferential level dependent on the problem.

Finally, we note that the knowledge levels distinguished correspond to different types of systems for spatial decision making. A general-purpose GIS provides tools that must be combined to solve application specific problems. Therefore, roughly speaking, these systems provide support up to the inferential level. A dedicated GIS provides in addition application-specific procedures possibly in the form of macros. Therefore, one might say that these systems provide support up to the task level. Finally, a spatial DSS adds to a dedicated GIS facilities for guiding decisionmaking and thus provides support up to the strategic level. In this scheme an expert system stands at the same level as a DSS with as a discriminating feature the incorporation of domain-specific strategic knowledge that controls rather than guides the decision making process.

**THE ROLE OF EXPERT SYSTEMS**

In this section we investigate the possibilities of applying ES techniques for enhancing the reasoning capabilities and intelligence of DSS at the inferential, task and strategic level. We start with describing an existing method for representing qualitative knowledge and we discuss its potentials for our purpose. Next, we discuss the possibilities of developing qualitative models at the different knowledge levels.
### FIGURE 4. Example of a Decision Table: The Viability of Neighbourhood Shopping Centres in a Dutch Context.

<table>
<thead>
<tr>
<th></th>
<th>&lt; 4500</th>
<th>4500 - 6000</th>
<th>6000 - 7500</th>
<th>&gt; 7500</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1 population in neighbourhood</td>
<td>-</td>
<td>strong</td>
<td>moderate OR weak</td>
<td>weak</td>
</tr>
<tr>
<td>C2 orientation of neighbourhood population to centre</td>
<td>-</td>
<td>strong</td>
<td>moderate OR strong</td>
<td>weak</td>
</tr>
<tr>
<td>A1 centre is viable</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>A2 highlight object</td>
<td>X</td>
<td>-</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>R1</td>
<td>R2</td>
<td>R3</td>
<td>R4</td>
<td>R5</td>
</tr>
</tbody>
</table>

### FIGURE 5. Example of a Decision Subtable: The Definition of Condition C1 of Decision Table 4.

<table>
<thead>
<tr>
<th></th>
<th>good</th>
<th>moderate</th>
<th>bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1 quality of centre</td>
<td>-</td>
<td>isolated</td>
<td>isolated OR mod. clust.</td>
</tr>
<tr>
<td>C2 position of centre in retail structure</td>
<td>-</td>
<td>central</td>
<td>ex-cent.</td>
</tr>
<tr>
<td>C3 position of centre in neighbourhood</td>
<td>-</td>
<td>ex-cent.</td>
<td>ex-cent.</td>
</tr>
<tr>
<td>A1 orientation of neighbourhood population to centre</td>
<td>strong</td>
<td>strong</td>
<td>weak</td>
</tr>
<tr>
<td>R1</td>
<td>R2</td>
<td>R3</td>
<td>R4</td>
</tr>
</tbody>
</table>

### Representing knowledge in decision tables

Decision tables were introduced over three decades ago primarily in the implementation stage of software engineering. Only recently the focus has shifted to using this technique for modelling and representing knowledge in knowledge-based systems. Compared to alternative methods of knowledge representation, such as logic, production rules and frames, the method has clear advantages regarding knowledge acquisition, structuring and validation. Lucardie (1994) describes a comprehensive study on the theoretical foundation of decision tables and the possibilities the method offers for building knowledge-based systems.

Fig. 4 and Fig. 5 illustrate the structure of decision tables. A decision table has a condition section (the upper part) and an action section (the lower part). The application domain of a table consists of all objects (e.g. entities, processes, states, events etc.) that are describable in terms of the set of condition subjects listed in the upper left section. The aim of a table is to assign a given object that belong to its application domain to an action described in terms of the action subject(s) listed in the lower left part. For example, the table of Fig. 4 assigns neighbourhood shopping centres to viability classes (yes or no) dependent on the population in the neighbourhood concerned and its orientation to the centre. Typically, an
action is a statement (conclusion) about an object, but it may also involve a procedure for performing some task or a combination of the two. The example shows the combination of a conclusion (A1) and a procedure (A2: assigning the highlight status to an object). Every table column represents a conditional statement (an if-then rule). For example, column R2 of the example table corresponds to the rule "if the population is within the range [4,500–6,000] and the orientation is strong, then the centre is viable". Therefore, a table can be thought of as a group of if-then rules that apply to the same domain and define the same type of action.

In a (well-formed) decision table the condition alternatives of a condition subject are mutually exclusive and cover the whole condition space (all possible states), so that every object is assigned to no more than one and at least one table entry. To return to the example, the alternative population categories exclude each other and cover all possible population figures so that every possible population size is uniquely assigned to an entry. These properties of decision tables are referred to as exclusiveness and exhaustiveness (coverage).

To find the appropriate table entry, an object must be classified on each of the relevant condition subjects. If a condition is formulated in abstract terms (e.g. population orientation) so that deciding on the entry (strong, moderate or weak) is not straightforward, the way this must be done can be defined in a so-called subtable. For example, the subtable in Fig. 5 defines the classification on orientation to own centre, which is a condition (C2) of the table in Fig. 4. A knowledge model typically consists of a head table that selects an action dependent on abstract conditions which are further specified in subtables and so on. Following this strategy, a knowledge model has the form of a hierarchy of tables. The end tables of the tree structure are formulated in terms that allow a direct matching with observable object attributes.

The usefulness of decision tables

Compared to if-then rules, decision tables impose a higher degree of structure to knowledge, which offers a number of advantages. First, decision tables are particularly suitable for classification tasks, because they facilitate the representation of complex interactions between classification factors (conditions). First, tables explicitly deal with the phenomenon that the relevance of factors for classification may depend on other factors. For example, in the table of Fig. 4, population's orientation (C1) is relevant for determining viability only if the population is in the range [4,500–7,500]. Furthermore, decision tables explicitly express the phenomenon that the categorisation of a factor relevant for the classification may be conditional on other factors, e.g. in the example, categories of population's orientation relevant for determining viability depends on the range, [4,500–6,000] or [6,000–7,500], within which the population (C1) falls. Finally, decision tables highlight the phenomenon that objects with different attribute profiles may belong to the same class implying that they are equivalent for the classification purpose. In the example (Fig. 4) this so called functional equivalence is reflected in that different population–orientation combinations may lead to the same viability classification (e.g. R2, R4 and R6).

Second, decision tables allow the model builder to check whether the knowledge represented is consistent and complete, simply by verifying the exclusiveness and completeness of condition alternatives, as explained above. In this way, one can make sure that the model behaves in a reliable and predictable way.

Third, decision tables not only facilitate the structuring of knowledge, but are also helpful in the previous stage of knowledge acquisition. They facilitate knowledge transfer between domain expert and knowledge engineer for several reasons. First, the tables represent complex knowledge in a form intelligible to lay people. Second, experts are able to check the
correctness of a model in a systematic way by verifying table columns (conditional statements) separately. Third, decision tables support a hierarchical organization of knowledge that provides modularity and helps to keep overview. The topdown approach of starting with an abstract table and recursively working out details corresponds with the way people tend to organize or transfer knowledge.

Finally, the software AKTS (Advanced Knowledge Transfer System) is available to develop and simulate models based on decision tables (Lucardie, 1994). AKTS supports the creation and modification of decision tables and provides an inference mechanism for consulting the knowledge base. The user is able to assign properties to (condition and action) parameters, such as a goal status, explanation texts, user prompts (text or images), and if-needed procedures (e.g. read from a database). Procedures can be built in either directly by entering Prolog source code or indirectly by a call to an external procedure (e.g. written in a procedural language). In a consultation session, the inference engine evaluates the parameter that has the goal status (usually, the action parameter of the head table). The relevant condition parameters are evaluated by executing the attached if-needed procedure (if any), evaluating the corresponding subtable (if any) or by asking the user. Usually, conditions of the goal table are recursively defined in subtables so that the value of the goal parameter is eventually found through a (traceable) backward chaining process.

In sum, decision tables provide a technique for the acquisition and modelling of complex qualitative knowledge. Particularly, the communicative properties, the possibility of knowledge validation and the availability of a system shell (AKTS) motivates our choice for this technique.

The application of expert systems

Decision tables or other knowledge representation techniques have the potential to improve the effectiveness of DSS in several ways. First, by providing solutions to subproblems for which no algebraic or algorithmic solutions exist. Second, by integrating task knowledge in existing algorithms to improve their intelligence and, third, by incorporating analytic and strategic knowledge for applying available methods and procedures to a given problem. This section investigates the applicability of expert system technology for these purposes.

Inferential level

The inference of system states or events from existing ones often involves qualitative reasoning. Therefore, ES technology might be particularly useful at the inferential level. Inference can be approached as the problem of factor combination, i.e. assessing a factor (e.g. a C-variable) based on one or more dependent factors (e.g. a set of X-variables). Symbolic knowledge rules are capable of dealing with complex dependencies between factors (Wright, 1990). Therefore, they might be useful if dependencies exist which can not be expressed by mathematical equations.

Decision tables deal with conditional relevance and conceptual interactions explicitly, as we have shown, and are therefore particularly suitable for expressing the complex relations between variables of linked system perspectives. In our approach, decision tables (or hierarchies of tables) are used as methods attached to variables, which classify objects on that variable (the action parameter) conditional upon variables (condition parameters) of linked views.

We have distinguished conceptual, causal and instrumental inference. In particular the conceptual type of inference may be suitable for ES applications. In planning problems, as explained before, conceptual inference is involved in the analysis and evaluation of system
The integration of expert knowledge in DSS for facility location planning

### Accessibility indicator

<table>
<thead>
<tr>
<th>C1</th>
<th>single</th>
<th>multiple</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
<td>single</td>
<td>multiple</td>
</tr>
<tr>
<td>C3</td>
<td>simple</td>
<td>complex</td>
</tr>
<tr>
<td>C4</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>C5</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A1</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
<th>D4</th>
<th>D5</th>
<th>D6</th>
<th>D7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
<td>R2</td>
<td>R3</td>
<td>R4</td>
<td>R5</td>
<td>R6</td>
<td>R7</td>
</tr>
</tbody>
</table>

Where:
- **D1** distance to nearest facility;
- **D2** distance to reach X-th facility destination of the same type;
- **D3** sum of available facilities of the same type attenuated with distance decay;
- **D4** some measure of the net utility derived from available facilities of the same type;
- **D5** length of a multistop trip connecting X facility destinations of the same type;
- **D6** distance to reach X facility destinations of different types;
- **D7** length of a multipurpose trip connecting X facility destinations of different types.

**FIGURE 6.** Example of Representing Analytic Knowledge in Decision Tables.

performance (deriving the Future State in C and the Discrepancy State respectively). In ES literature, this type of task is known as interpretation tasks. As reviewed by Ortolano and Perman (1990), there have been numerous expert system applications for interpretation tasks, which evidences the potential applicability of expert systems to this type of problems. Particularly, many examples of ES for site-suitability analysis exist (e.g. Findikaki, 1990; Suh et al., 1990; Han & Kim, 1990b; Rouhani & Kangari, 1990), which suggests that this conceptual subproblem of spatial planning is suitable for an ES approach. The assessment of the viability of neighbourhood centres discussed above is an example of a conceptual inference task that can be formalized using ES.

Probably, the usefulness of ES for causal and instrumental inference, on the other hand, will be limited. Causal inference, i.e. forecasting and impact assessment, is dominantly a data rather than knowledge intensive task and can often be approached by statistic projection and modelling techniques. Instrumental inference, i.e., deriving actions from discrepancies, often involves common sense reasoning which is beyond the capability of ES (Prerau, 1989).

**Task level**

At the task level we have made a distinction between analytic and task procedural knowledge. The analytic task involves the selection (and, if needed, the specification) of inferential methods dependent on information needs and available data resources and, furthermore, the interpretation of obtained results in a problem-oriented way. ES may play a significant role in performing this task as evidenced by many KBDSS described in the literature (e.g. Krishnan, 1989; McGovern & Samson, 1989; Elam & Konsynski, 1987; Moser, 1986). The decision table in Fig. 6 illustrates how analytic knowledge can be represented in decision ta-
bles and the type of function it may perform. The goal of the table is to specify an indicator of the accessibility of facilities dependent on the information needs of the planner. Complementary tables could be specified for the selection of appropriate X-variables (e.g. the way of measuring demand weights, supply size, distances etc.) dependent on the method chosen and available data sources. Tables of this kind assist planners in the specification of system perspectives.

At the level of task procedures, ES can be used to incorporate domain-specific knowledge in generally applicable algorithms to improve the efficiency or flexibility of task procedures. ES has proven to be useful for reducing in an intelligent way the search space of combinatorial algorithms (e.g., Erickson et al., 1989). Armstrong et al. (1990) describe a strategy in which ES is used in combination with a location-allocation algorithm. They suggest to use heuristic rules for reducing the problem size (the data input) and improving the computational efficiency (during execution) of algorithms.

**Strategic level**

In contrast to the task layer, problems at the strategic layer are ill- or semi-structured and require, therefore, flexible problem solving. If all the relevant strategic knowledge could be formalized, an ES solution to the facility planning problem would exist. However, earlier we have argued that no such solution exists. Armstrong et al. (1990) and Wyatt (1991) have shown that ES can be used instead for assisting planners in developing a strategic plan for achieving a given goal. Furthermore, at this level ES could support the monitoring, diagnosing and repairing of the progress in the problem solving process (Breuker & Wielinga, 1989). Strategic knowledge developed in spatial planning research can be represented in scripts. A script specifies (well-defined) steps for achieving a certain goal and suggests alternative strategies when steps do not yield the intended results. As an example, Fig. 7 shows a script for the selection of locations with high trading potential (possibly, for expanding a retail chain). The steps correspond to goals that can be achieved by standard available task procedures. The flexibility of a script is reflected in the fact that (some of) the steps are conditional upon obtained results. The statements included are not internal commands to execute procedures, but rather guidelines presented to users for structuring the problem. In this example the script is independent on internal (e.g. available data) and external constraints (e.g. preferences of the decision maker). However, in other cases steps could be made dependent on these problem characteristics. Then, a strategic plan would be generated in interaction with the user. The scripts proposed here are very similar to metaplans in the KBDSS described by Armstrong et al. (1990).

It can be concluded that in a DSS context ES seems to be particularly useful for interpretation tasks (conceptual inference), interfacing the model base (analytic knowledge), improving the intelligence of algorithms (task knowledge) and supporting the structuring of the problem (strategic knowledge).

**A KNOWLEDGE-BASED DSS**

This section focuses on the place and role of the ES components identified above in a KBDSS for facility planning. We assume a view-based DSS that we have developed in a former study (1994b) and that is currently being implemented. The system can be used at all four problem levels. The system levels are hierarchical in that a higher level has all the functionality of lower levels and more. We now briefly discuss the use of the system and the role of ES at the different levels.
<table>
<thead>
<tr>
<th>1. Do</th>
<th>&quot;identify candidate market regions&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arguments:</td>
<td>type of branch</td>
</tr>
<tr>
<td>If</td>
<td>no market regions are found</td>
</tr>
<tr>
<td>Then</td>
<td>Reconsider: type of branch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. Do</th>
<th>&quot;select market region&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arguments:</td>
<td>candidate market regions (choice set)</td>
</tr>
<tr>
<td></td>
<td>economic base and LT perspective (criterion)</td>
</tr>
<tr>
<td></td>
<td>population base and LT perspective (criterion)</td>
</tr>
<tr>
<td>If</td>
<td>no market region selected</td>
</tr>
<tr>
<td>Then</td>
<td>Reconsider: criteria OR step 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. Do</th>
<th>&quot;delineate homogeneous market sectors&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arguments:</td>
<td>selected market region (focus area)</td>
</tr>
<tr>
<td></td>
<td>physical barriers (criterion)</td>
</tr>
<tr>
<td></td>
<td>movement pattern of people (criterion)</td>
</tr>
<tr>
<td></td>
<td>demographic profile of population (criterion)</td>
</tr>
<tr>
<td>If</td>
<td>no market sectors are delineated</td>
</tr>
<tr>
<td>Then</td>
<td>consider market region as market sector</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4. Do</th>
<th>&quot;select suitable market sectors&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arguments:</td>
<td>type of branch</td>
</tr>
<tr>
<td></td>
<td>retail potential (criterion)</td>
</tr>
<tr>
<td></td>
<td>competitive strength (criterion)</td>
</tr>
<tr>
<td>If</td>
<td>no market sector selected</td>
</tr>
<tr>
<td>Then</td>
<td>Reconsider: selection of market region</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5. Do</th>
<th>&quot;identify candidate locations&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arguments:</td>
<td>market sectors (focus areas)</td>
</tr>
<tr>
<td></td>
<td>screening criteria (criteria)</td>
</tr>
<tr>
<td>If</td>
<td>no candidate locations found</td>
</tr>
<tr>
<td>Then</td>
<td>Reconsider: previous steps</td>
</tr>
<tr>
<td>If</td>
<td>set of candidate locations too large</td>
</tr>
<tr>
<td>Then</td>
<td>Consider: narrowing down the selection of market sectors</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6. Do</th>
<th>&quot;select suitable sites&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arguments:</td>
<td>candidate locations (choice set)</td>
</tr>
<tr>
<td></td>
<td>trading potential (criterion)</td>
</tr>
<tr>
<td></td>
<td>marketing mix</td>
</tr>
<tr>
<td>If</td>
<td>no suitable sites found</td>
</tr>
<tr>
<td>Then</td>
<td>Reconsider: selection of market region OR market sectors OR marketing mix</td>
</tr>
</tbody>
</table>

**FIGURE 7. Example of a Strategic Plan.**

**The domain level**

The view-based DSS provides users with a standard set of views that correspond to the different system perspectives in the scheme of Fig. 1. A view is a frame-like structure wherein users can specify variables and attached methods (available in the method base of the system). At the domain level, the system is used as a database system. The user defines a data model appropriate to the given location problem by specifying variables (data fields) in the views of interest. The corresponding data can be entered or imported from existing data.
bases (e.g. a GIS database). Thus, the views are different cross sections of the database that show different perspectives of the facility system being investigated. At this level, the system further supports the editing of data, e.g. to implement changes in assumed autonomous developments, action plans or goal formulations. The data within views can be edited and displayed in table, map or diagram format. For example, a map interface can be used for locating or relocating entities in the study area and for displaying thematic information (view-variables). In sum, at the domain level the system is used for defining the database structure (view variables) and to store, edit and display data in views.

The role of ES at this level is to support users in choosing the variables (data fields) of interest. Based on analytic knowledge, the system infers from a given goal state (possibly stated in informal terms) possibly relevant C, X and I-variables and presents these variable dimensions to the user. Furthermore, a library of standard data models is available. The user can select a model that suits the problem and next adapt the model to specific information needs.

**Inferential level**

At this level, a library of inferential methods is available that can be attached to variables in (dependent) views. Included are methods for forecasting, impact assessment (Future State in X), performance analysis (Future State in C), performance evaluation (Discrepancy State) and plan generation (Actions in X or I). The user can select available methods from view-specific lists presented by the system. Dependent on the method selected, the system starts a dialogue for setting parameters, if any, and selecting variables of dependent views that serve as data input. Once variables and methods are specified, the user can interact with the views to solve a planning problem. Typically, the user implements different scenario's by changing action plans (decision alternatives), goal variables (possible goal formulations) and autonomous developments (possible base scenario's). An inference engine is built in that automatically updates view variables upon actions of users by triggering attached methods. Thus, users can simply activate a linked view of interest (typically, the Discrepancy State view) to obtain feedback information on actions (typically, changes in the Action view). Besides information on view variables, users can obtain: (i) views on method specifications; (ii) a report of the procedure used to evaluate (update) a variable; and (iii) a report of the analysis carried out. An analysis report is simply a transcript of information contained in views, which shows not only analysis results (variables of dependent views) but also the methods (attached methods) and data (variables of independent views) that were used. In sum, at the inferential level the system is primarily used for performing what-if types of analysis. An action plan is developed through an iterative process of adjusting and evaluating possible actions (including goals and autonomous developments).

At this level, the role of ES is twofold. First, inferential knowledge, particularly for interpretation tasks, is incorporated in the form of knowledge-based methods that can be attached to variables. Second, analytic knowledge is built in for supporting not only the specification of view variables (as at the domain level) but also the selection of appropriate attached methods. Furthermore, this type of knowledge is used for the generation of procedure reports that explain results in non-technical terms. Finally, the system provides a library of predefined (prototypical) view-based models. Users can select a prototype model and next adapt the model to specific information needs. Reversely, view-based models developed by the user can be added to this library for reuse.
Task level

At the task level, the system has a library of algorithms for performing well-defined tasks, such as, for example, the selection of candidate locations, the generation of plan alternatives and the ranking of plan alternatives. At this level the user solves a given location problem by selecting standard procedures for solving subproblems. A procedure defines a view-based model (variables and attached methods) and a sequence of inference steps. If the procedure selected leaves room for choosing between different method specifications, the system starts a dialogue to define settings. For example, in the case of a location–allocation algorithm for location selection, the system asks the user to specify methods for evaluating C-variables (the objective function) and X-variables (the allocation function, demand weights etc.). Of course, the functionality at the inferential level is also available so that the user can switch between the task procedural and inferential level. For example, having run a procedure for generating an action plan, the user may switch to the inferential level to perform what-if analyses (e.g. simulating possible autonomous developments to assess the robustness of the action generated). The user may define new task procedures in the form of macros of inferential steps, which can be added to the task library for reuse. At the task level, ES plays the role of improving the intelligence of task procedures by incorporating domain-specific knowledge, as explained before.

Strategic level

At the strategic level, users interact with the system to develop a strategy for achieving a given goal. The system provides a library of standard strategies in the form of scripts that define a sequence of procedures to achieve a certain goal. Dependent on the goal specified, the system selects a script or, if no such script exists, brings to bear generic strategic knowledge to help users to structure the problem. In interaction with users and dependent on preferences, available data and other constraints, task procedures are selected. Of course, users are free to switch to lower levels to execute standard procedures, perform what-if analysis or view data. At any stage in the process, users can return to the strategic level to receive guidelines on how to proceed.

The system provides one integrated environment that does not change when users move from one level to another. A level is simply defined by the way users interact with the system, i.e. the type of function chosen (from one general menu structure). Internally, the system levels are reflected in the different libraries of data models, inferential methods, analytic models, macros and scripts the system provides. To summarize, a data model is a predefined structure of view-variables, an analytic model is a predefined structure of variables and methods (an inference structure) that describe a facility system, a macro is a preprogrammed sequence of methods that defines a task procedure and a script is a flexible sequence of task procedures that defines a strategy. The DSS not only provides libraries at each of this levels, but also enables users to define and store new data models, analytic models or macros. Expert knowledge is incorporated in each of these layers to enhance the modelling capabilities and intelligence of the DSS at the different levels.

CONCLUSIONS AND DISCUSSION

In this paper we investigated the application of expert system techniques for improving the modelling capabilities of DSS for facility planning. The knowledge involved in facility planning can be described by a four-layer model. The domain and inferential layers describe the facility system under concern in terms of states and events and the way they are related.
The subsequent task and strategic layers prescribe the way planning problems should be solved at the level of well-defined subproblems and the ill-structured overall problem. Every higher level layer controls (or constraints) the processes that can take place one level lower.

ES techniques make it possible to incorporate expert knowledge for complementing existing algebraic and algorithmic models in a DSS. Decision tables are particularly useful for representing complex qualitative knowledge. This formalism facilitates knowledge acquisition, the representation of complex factor interdependencies and knowledge validation. ES techniques are particularly useful for developing conceptual methods at the inferential level and flexible and intelligent procedures at the task level. Furthermore, analytic and strategic knowledge can be incorporated to assist in the use of available methods and in the structuring of the planning problem.

The DSS we are developing uses a standard set of active and linked views (perspectives) on the facility system being investigated. The DSS can be used and provides support at four levels: editing and viewing of data (domain level), performing what-if analyses (inferential level), applying standard solution procedures (task level) and developing solution strategies (strategic level). Expert knowledge is included in each of the three procedural levels in the form of conceptual inferential methods, intelligent task procedures and flexible strategies. Therefore, it provides a suitable framework for using different forms of knowledge, domain and meta knowledge, as well as different modelling techniques, algebraic, algorithmic and ES, in combination. The resulting system is highly flexible and provides intelligent support at all four levels of planning problems.

ACKNOWLEDGEMENTS—This research is supported by the Technology Foundation (STW).

REFERENCES


The integration of expert knowledge in DSS for facility location planning


