The Korean Journal of Regional Science, Vol. 6(2)
December 1990, pp. 169~180

The Application of Knowledge-Based Decision Support System in Regional Science*

Sang-Yun Han

Department of Public Administrations, Kyung Hee University

Tschangho John Kim

Department of Urban and Regional Planning, University of Illinois at Urbana-Champaign

1. Introduction

Decision makers in both the public and private sectors often face the difficult problem of effectively managing facilities that range from utility plants to small offices. The task of facility management not only requires quantitative reasoning but also a great deal of qualitative reasoning. In developing the best possible facility management plan, one should have easy access to a great amount of facility inventory data, knowledge of experts on various domains including diagnosis of facility condition and analytical mathematical models to ensure efficient use of resources.

Despite the fact that separate approaches can provide only partial solutions to the problem, there has been no attempt to devise an *integrated* computer tool in the area of facility planning and management. Recent advances in artificial intelligence (AI), particularly expert systems (ES), can provide a means of narrowing the gap between what traditional problem-solving methods can do and what decision makers want (Han and Kim 1989)[2].

To shed light on how to combine traditional programming models and expert systems, this paper has the following objectives: (1) to briefly review how an operations research (OR) approach to problem solving is different from an AI approach, (2) to discuss how to develop an integrated computer tool which effectively aids decision-making in facility planning and management. For the second objective, this paper presents the design and implementation plan of a knowledge-based decision support system XPlanner. XPlanner combines ES with a zero-one integer optimization model and a database system to create a comprehensive decision aid for the management and planning of military facilities.

2. Operations Research and Artificial Intelligence

The field of OR is large and includes many subareas. The problems that have been solved by OR include queueing, inventory, allocation, routing, scheduling, search, replacement and competition (Wilson 1985)[15]. AI, on the other hand, has focused on such areas as representation and use of human knowledge through logic, learning and understanding of natural language and perception (vision and touch). Simon (1987)[11] compares OR with AI by defining OR as "the application of optimization techniques to the solution of complex problems that can be expressed in real numbers" while defining AI as "the application of methods of heuristic search to the solution of complex problems that defy the mathematics of optimization, contain nonquantifiable components and involve a large knowledge base."

Both expert systems (ES) and OR models

*Paper submitted for possible publication in the *Journal of Korean Regional Science Association*.

Table 1. Different Approaches of Linear Programming Models and Expert Systems

	LP models	Expert systems
Knowledge	Constraint equations	Knowledge base: rules and frames
Solution method	Simplex algorithm	Inference engine: backward and forward chaining

are designed to help decision makers. But while ES normally incorporates qualitative knowledge, OR primarily uses quantitative knowledge. Table 1 briefly compares the different solution approaches of linear programming (LP) models and ES. The linear programming model, for instance, involves arithmetic computations to solve a goal function:

Minimize $\Sigma_i \alpha_i X_i$

subject to $\Sigma_i \beta_i X_i \leq A$.

Using a simplex algorithm, it finds the optimum values for the objective function (goal).

On the other hand, ES involves symbolic processing in representing knowledge and finding a value of goal parameter. For a simple example of evaluating land development proposal, consider the following knowledge base written in Prolog syntax:

Goal *acceptability (X, development-proposal)*

Rule 1 *zoning-change (needed, X): landuse (incompatible, X)*

Rule *acceptability (reject, X): zoning-change (needed, X)*

Fact *landuse (incompatible, development-proposal)*

In finding the value of the goal (e.g., acceptability of the land development proposal), ES may use backward chaining (deduction) to find out if the proposal requires a zoning change.[1] This is a simple example of the knowledge base and inference engine. Actual problems normally consist of hundreds of rules and a long inference process.

As shortly reviewed here, OR and AI (including ES) have developed different solution techniques, according to the type of knowledge they possess. The important question here is how and when OR and AI could be combined to take advantage of both their strengths. The rationale for the integration of these two fields is that most decision-making problems, particularly in urban and regional planning, require both quantitative and qualitative knowledge and reasoning. The facility management problem, for instance, may be quickly and optimally solved by OR techniques once the nature of the problem is mathematically formulated. But the process of formally defining the problem requires significant expertise in many subject areas of facility management. In addition, implementing the solution provided by the OR models may be better handled by ES through its qualitative reasoning process. The benefits of coupling ES and OR in this regard will be evident as the design concepts of XPlanner are discussed later.

3. Expert Systems and Decision Support Systems

Although some authors see expert systems as special instances of decision support systems (DSS), many authors distinguish ES from DSS because of the former's unique structure, technology and ability to solve different types of problems (see, for instance, Kroeber and Watson 1987[4]; Liang 1988[7]). Table 2 summarizes major differences between ES and DSS in terms of types of computer processing involved and types of decision problems solved.

In DSS, the computer is used to store data and various decision models and the user interacts with the computer throught the user-interface, providing the computer with judg-

Table 2. Taxonomy of Decision Problems

Type of knowledge needed for problem solving	Nature of problems	
	Structured or routine	Unstructured or semistructured
Quantitative reasoning: numeric computation	Data processing systems	Decision support systems
Qualitative reasoning: intuition and rules of thumb	Expert systems	Human experts

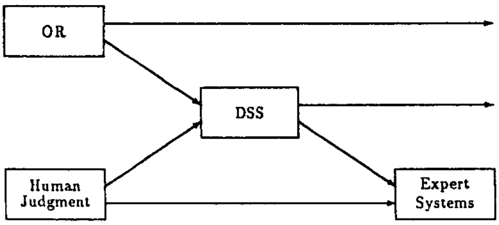

Figure 1. Progression of Operations Research to Decision Support Systems and Expert Systems (Tingley 1987)

ments. In ES, the computer stores all expert knowledge, including judgments, into a program and recommends a solution by using appropriate analyses and its own programmed logic.

In discussing the possible relationship between OR, DSS and ES, Wynne (1984)[16] and Tingley (1987)[13] describe these techniques as a sequence of means for providing assistance to decision makers (Figure 1). OR models are developed into DSS to incorporate human judgment into the decision-making process through an effectively developed man-machine interface. DSS, in turn, can be further developed into ES, in which human judgment is coded into a knowledge base. The feasibility and desirability of these progressions from OR to DSS and from DSS to ES are, of course, dependent on the nature of decision problems.

The idea of integrating ES into DSS to create more powerful and useful computer-based systems has recently garnered much attention, creating a new terminology, "expert decision support system" (DESS), "knowledge-based decision support system," or "intelligent decision support system." The possible contributions of ES to EDSS include: (1) helping users in selecting models, (2) providing judgmental elements in models, (3) simplifying building simulation models, (4) enabling friendlier interface and (5) providing explanation capability (Turban and Watkins 1986) [14].

Of these, the most important role of ES in EDSS is in model selection and building. As Strauch (1974)[12] points out, the process of problem analysis (or modeling) usually involves three interrelated components: formulation of the formal problem, mathematical analysis and interpretation of the results. While the mathematical analysis is handled effectively with DSS through its embedded OR models, the formulation requires the subjective knowledge of the user. Further, the interpretation requires the personal judgment of the decision makers. The coupling of ES and DSS in this case is based on the assumption that subjective knowledge and personal judgment can be better made by experts than by decision makers and users of the system.

The sections that follow discuss the design concepts and implementation plan of XPlanner. XPlanner is an example of adding a rule-based system to the optimization model for facility management for the purpose of aiding users with the various tasks involved, such as formulating facility optimization models and diagnosing structural conditions of facilities. It is developed to stimulate planners to employ mathematical models more frequently and easily in their problem-solving processes.

4. Decision Problems Supported by XPlanner

Most decision problems may be broadly categorized into (1) structured and (2) unstructured, or semistructured. XPlanner is developed based on the notion that there are different types of computer-based information systems, each with its own unique ability to solve decision problems. Data processing systems are suited for structured problems that have standard operational procedures, decision rules and clear output formats, such as identifying low income districts or determining the median income of a city. Decision support systems (DSS), on the other hand, are intended for unstructured or semistructured problems, such as evaluating land development proposals, for which DSS can be used to perform "What-If" type analyses estimating fiscal and other impacts of proposals based on different sets of variables, providing quantitative support to the decision maker. Kroeber and Watson (1987)[4] define DSS as "an interactive system that provides the user with easy access to decision models and data in order to support semistructured and unstructured decision-making tasks." As the definition implies, the interaction between the decision maker and the system is very important in DSS. The interaction is usually achieved in the form of "What-If" dialogue.

While expert systems (ES) are very good at solving problems which require qualitative reasoning, they have the strict requirement that the decision problem be structured so that experts' knowledge on solving the problem can be captured in a computer program. As the characteristics of each computer system imply, only the integration of different types of computer-based systems can produce effective decision-making aids because most decision making in urban planning deals with a mixture of structured and unstructured problems.

The task of facility planning and management has both the structured and unstrunctured decision-making components. Cities in urban areas and military installations maintain a variety of facilities from utility plants to housing units and the decision tasks involved in facility management range from the task of determining the current physical and functional conditions of facilities to that of deciding which facilities to close or build. While the former example is a somewhat structured decision problem, the latter example can be regarded as an unstructured or semistructured problem.

XPlanner is targeted for both areas of decision making. In military installations, for which XPlanner is developed, the decision problems in facility planning and management can be categorized as follows:

1. Project possible changes in planning constraints such as mission types and budget levels.
2. Estimate facility requirements based on the missions of installation.
3. Determine the current physical and functional conditions of facilities.
4. Estimate facility deficiencies based on 2 and 3 above.
5. To handle the facility deficiency or surplus, create a facility management plan for an effective and efficient utilization of existing and future resources.

Only the effectively integrated computer system can support all the decision-making tasks listed above. The task of projecting mission and budget changes can be handled by an effective man-machine interface of DSS, which incorporates the judgments of the decision maker and enables a series of "What-If" dialogue. The "What-If" interaction is critical in developing XPlanner, because decision--makers want to incorporate their judgment on the future of the Army (e.g., mission and budget changes) into the problem solution process of XPlanner. For example, when the user inputs different types of missions, the system produces new facility management plans.

The task of determining current *physical* conditions of facilities can be handled by a database management system that maintains comprehensive facility inventory data, whereas the task of determining *functional* conditions of facilities may better be handled by an expert system. Examining the functional conditions of facilities involves such heuristic tasks as determining whether a particular facility is suitable for accommodating a certain type of activity and assigning each activity (e.g., training or recreation) to a facility. Further, the task of developing a facility management plan can be supported by an optimization model in order to achieve efficient allocation of resources.

In short, the major decision problem involved in military facility planning and management is developing an effective facility management plan that supports constantly changing missions. The facility management plan prepared by the Army planners should effectively utilize the current and future resources of the military installation and should also comply with the safety and welfare standards prescribed by the Army authorities. XPlanner is designed to support these tasks.

5. Structure and Components of XPlanner

To effectively support all tasks involved in facility planning and management, XPlanner consists of several components (Figure 2): (1) knowledge base, (2) model base, (3) database, (4) user interface and (5) inference engine. The inference engine and knowledge base control the whole system.

1) The Role of the Knowledge Base

The knowledge base of XPlanner contains two types of knowledge: (1) knowledge about the classification of functional areas of facilities and about the diagnosis of the physical conditions of facilities and (2) knowledge of the formulation and interpretation of a zero-one integer optimization model for facility planning management.

For the examples of the first type of knowledge, consider the following rules used in XPlanner for diagnosing physical problems of structure:

1. IF structure = wood-frame AND humidity-level >= 80, THEN termite-infestation may be high.
2. IF termite-infestation = high, THEN building-decay = serious AND wood-treatment by chemical A = required.
3. IF sagging-roof OR non-vertical-walls OR tilted-floors OR misalignment-of-doors, THEN excessive-settling.
4. IF excessive-settling, THEN foundation-wall = to-be-replaced.

These rules can be used to effectively estimate the cost and feasibility of each type of project (e.g., conversion and renovation) related to each facility. Based on this type of knowledge, the major function of the knowledge base is to estimate parameter values necessary to formulate the zero-one integer optimization model. The list of parameters to be estimated by ES (the knowledge base component) is described later under the model base.

In addition to estimating parameter values for the facility optimization model, ES screens existing facilities in order to eliminate the facilities that are irrelevant for consideration by the optimization model. For this purpose, ES maintains the knowledge base that classifies existing facilities into several condition categories ranging from Class 1, usable (meeting all criteria), to Class 6, disposable (no longer tenable for any purpose). Machine learning programs with effective induction algorithms can be used to develop the knowledge base in this area. Many induction algorithms have been successfully applied to the knowledge acquisition problems, particularly to classification problems (Michalski (1985) [8]; Forsyth and Rada (1986)[1]; Shaw (1988)[10]; Liang (1989)[6]). Because ES classifies and eliminates the irrelevant facilities, the facility optimization problem is reduced to a size manageable by the zero-one integer optimization model (described later).

Another important role of the knowledge base is to interpret the results of the optimization model using its knowledge of a zero-one integer model. In this system, ES serves as an extra layer between the model and the user, translating qualitative criteria into the numeric input and also translating the model's numeric output to qualitative concepts that are more intuitive and informative to the user.

In addition, the knowledge base has the important function of controlling the whole system, accessing the database and the model base of XPlanner as necessary. In addition to the domain knowledge specific to the facility management, the knowledge base maintains control knowledge (meta-knowledge) to control the steps involved in reasoning. For instance, it initiates the forward chaining mode in the middle of backward chaining[2] or it forces some rules to be fired before other rules by using meta-rules. This type of control knowledge is necessary to control the flow among the knowledge base, the model base and the database.

2) The Role of the Model Base

The model base of XPlanner contains a zero-one integer optimization model that is designed to devise a facility management plan with efficient allocation of resources. Integer programming (IP) is a special case of linear programming with the characteristic that the values for all the variables in the solution must be integers. A further sub-class of IP is zero-one programming, in which all variables in the solution have a value of zero or one. Such a formulation is needed when the decision variables represent a binary decision. An example of the set of binary decisions involved in XPlanner are *renovate* or *don't renovate*. IP problems are computationally much more difficult to solve than linear programming problems (Greenberg 1971)[3].

All the parameter values contained in this model are supplied by the knowledge base. In return, the model base supplies model output to the knowledge base for the interpretation (see Figure 2). The structure of the model is described here without further discussion on zero-one integer programming.

(1) Objective Function

The objective function is to minimize troop stationing facility costs, which include new construction, renovation, conversion and maintenance costs of facilities. To allow proper comparison of alternatives in equivalent dollars, both present and future costs for each

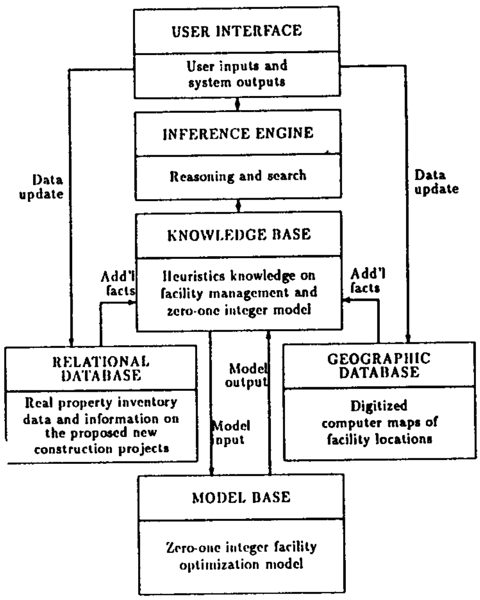

Figure 2. Schematic Structure and Components of XPlanner

alternative are brought to a common point in time by being converted to an annual series of payment. The objective function is formulated as below:

$$\text{Minimize } \sum[\mu_i^s S_i] + \sum_i[(\alpha_i + \mu_i^R)R_i] + \sum_i\sum_j[(\beta_{ij} + \mu_{ij}^C)C_{ij}] + \sum_l[(\gamma l + \mu_l^N)N_l] \quad (1)$$

where

$S_i =$ 1 if facility i is to stay as it is
0 otherwise

$R_i =$ 1 if facility i is to be renovated
0 otherwise

$C_{ij} =$ 1 if facility i is to be converted to j
0 otherwise

$N_l =$ 1 if new construction project l is to be approved
0 otherwise

$\alpha_i =$ annualized initial renovation costs of facility i

$$= a_i \left[\frac{r(1+r)^{n_i}}{(1+r)^{n_i} - 1} \right], \text{ where}$$

a_i = initial renovation costs of facility i,

r = discount rate and

n_i = economic life of facility i,

β_{ij} = annualized initial costs of converting facility i to facility j,

γ_l = annualized initial costs of new construction project l,

μ_i^S = annualized present value of total life time maintenance costs (recurring and non-recurring) of facility i when facility i stays as it is. Non-recurring cost is first converted to its present value before being annualized.

μ_i^R = annualized present value of total life time maintenance costs of facility i when facility i is renovated,

μ^C_{ij} = annualized present value of total life time maintenance costs of facility i when facility i is converted to j,

μ_l^N = annualized present value of total life time maintenance costs of new facility to be constructed by project l.

As shown above, the cost figures used in the objective function represent life cycle cost which is the sum of the initial, recurring and nonrecurring costs expressed in equivalent dollars. The values of the parameter, α, through μ_l^N, are endogenously determined by the knowledge base of XPlanner. When there are still facility deficiencies in the optimization model results, ES considers new construction (not already proposed) by checking the suitability of available space.

(2) Choice Constraints

The constraint below is to ensure that only one option (e.g., renovate, convert or use as it is) is selected for each existing facility:

$$S_i + R_i + C_{ij} = 1 \; \forall i, j \tag{2}$$

(3) Budget Constraints

The facility optimization model has the following budget constraints to ensure that the facility management plan is devised within given budgets:

$$\Sigma a_i R_i + \Sigma b_{ij} C_{ij} + \Sigma g l N_l < A \tag{3}$$

$$\Sigma(\mu_i^S S_i + \mu_i^R R_i) + \Sigma\Sigma \mu_{ij}^C C_{ij} + \Sigma \mu_l^N N_l < B \tag{4}$$

where:

a_i = total intial costs of renovating facility i.

b_{ij} = total intial costs of converting facility i to j.

gl = total intial costs of new construction project l.

A = total (one time) budget available for renovation, conversion and new construction projects.

B = annual budget available for minimal maintenance of facilities.

The knowledge base of XPlanner estimates the costs a_i, b_{ij} and g_l, and the users supply the value of A and B.

(4) Mission Constraints

The mission constraints are to ensure that the solution by the model satisfies the facility requirements generated by the mission Through the user interface, XPlanner aids users in developing scenarios regarding possibl mission changes. The knowledge base maintains the rules that interpret the scenarios developed by the user into the mission constraints (i.e. facility requirements). The mission constraints are formulated as below:

$$\Sigma(\delta^S_{ikp} S_i + \delta^R_{ikp} R_i) + \Sigma\Sigma \delta^C_{ijkp} C_{ij} + \Sigma \delta^N_{lkp} N_l > T_{kp}, \; \forall k, p \tag{5}$$

where:

δ^S_{ikp} = facility i's capacity of accommodating troop type k in activity p when facility i stays as it is.

δ^R_{ikp} = facility i's capacity of accommodating troop type k in activity p after renovation.

δ^C_{ijkp} = facility i's capacity of accommodating troop type k in activity p after conversion to j.

δ^N_{lkp} = new construction project l's capacity of accommodating troop type k in activity p.

T_{kp} = total number of troops of type k in activity p as given by new mission.

All the parameters in this model (e.g., α_i through δ^{N}_{lkp}) are estimated by the knowledge base. An example of the parameter supplied by the knowledge base is the capacity of facility i(i = 1,...., n) to accommodate troop type k(k = 1,...., enlisted woman,...., m) for activity p(p = 1,...., recreational use,...., l) after the facility is renovated. In addition, the values of the right-hand side constants (A, B and T_{kp}) are also supplied by the knowledge base.

The task of the ES component of XPlanner to configure the zero-one integer optimization model requires much computer power and speed. Even with eight facilities, the number of decision variables in the objective function is forty, which is the maximum that can be solved by the integer program package called Lindo. The number of parameters to be estimated by the knowledge base is 472, with eight facilities, three conversion types, three activity areas (training, recreation and housing) and four troop types (enlisted man, enlisted woman, noncommissioned officers and officers).

3) The Role of the Database

As an additional component, XPlanner maintains a relational database to store and access factual data necessary in facility management and planning. The database basically maintains four types of data:

1. Real property inventory data which describe existing conditions of facilities, such as area, current use, age, type of building materials used and structural conditions of each facility.
2. Detailed information on the new construction projects which have been proposed for the installation. It includes completion year, location, type and expected capacity of facilities.
3. Data on construction and repair materials. It includes material types, costs and durability.
4. Information on the space allowance standards compiled from the Army regulations and manuals.

In addition, XPlanner maintains a geographic database in which digitized computer maps are stored to provide spatial data such as location of roads and utilities (Figure 2). This information is used by the ES module of XPlanner in recommending candidate sites for new construction.

The knowledge base accesses the database to obtain additional facts. In XPlanner, the data flows from the database to the knowledge base and finally to the model base. The software dBase III Plus or the spreadsheet of Lotus 1-2-3 may be used as a database manager of XPlanner. The geographic database is developed by using Auto Cad and is utilized through a graphic utility program of the ES shell, Personal Consultant Plus.

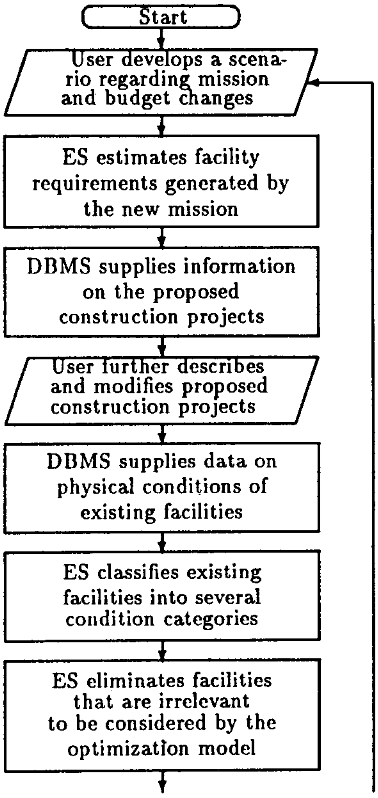

Figure 3. Flowchart of XPlanner (Part 1)

6. Workings of XPlanner

The workings of XPlanner involve frequent interactions among the user, the knowledge base, the model base and the database. The user provides XPlanner with additional judgmental factors not encoded in the knowledge base and the database supplies additional facts to the knowledge base. The knowledge base generates inputs for the model base and the model base, in return, supplies model outputs to the knowledge base (see Figure 2). As depicted in the flowchart of XPlanner in Figures 3 and 4, the steps involved in the consultation of XPlanner can be summarized as follows:

1. As the consultation with XPlanner begins, XPlanner first helps users develop their own scenarios regarding the future of the Army. They can play with their judgment in deciding possible mission changes, budget levels and some demographic changes (e.g., participation rate of woman labor force in the military).
2. ES of XPlanner estimates facility requirements generated by the new mission, using it knowledge about the space allocation standards set by Army regulations and manuals.

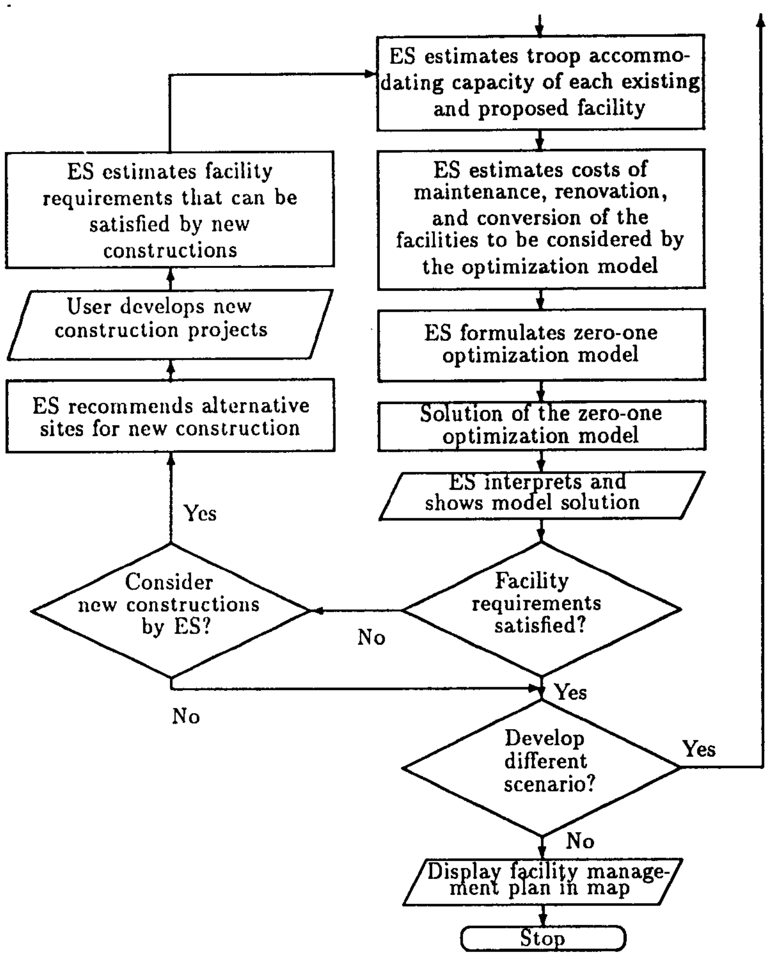

Figure 4. Flowchart of XPlanner (Part 2)

3. The database management system (DBMS) of XPlanner supplies information on the proposed construction projects and the users further describe and modify the nature of the projects if they want.
4. XPlanner accesses the database (real property inventory data) to check current physical conditions of facilities.
5. Based on the physical conditions, the ES module classifies existing facilities into several condition categories ranging from Class 1, usable (meeting all criteria) to Class 6, disposable (no longer tenable for any purpose).
6. ES eliminates the facilities that are irrelevant for consideration by the facility optimization model.[3]
7. ES estimates troop accommodating capacity of each existing and proposed facility. This information is needed later by the optimization model for its mission constraint equation (see Equation 5).
8. ES estimates costs of renovation, maintenance and conversion of the facilities to be considered by the optimization model, using its heuristic knowledge in the knowledge base. The costs of new construction are given by the project proposal.
9. After obtaining all the necessary parameter values, ES formulates the zero-one integer facility optimization model. As described earlier, the decision variables included in the model are whether to (1) renovate facility i, (2) convert facility i to j, (3) use facility i as it is and (4) approve new construction project l.
10. The facility optimization problem formulated in an integer programming model is solved.
11. The ES components of XPlanner interpret the results of the optimization model and explain the results to the user.
12. XPlanner asks the user if the results given by the model are acceptable in terms of facility requirements. If yes, go to Step 13; otherwise go to Step 15.
13. XPlanner asks the users if they want to develop another scenario. If yes, go to Step 1 and repeat the whole process; otherwise go to Step 14.
14. XPlanner displays the results (facility management plan) in computer graphic (map) format.
15. If the results given by the optimization model are not acceptable to the users in terms of facility requirements, XPlanner asks the users if they want XPlanner to develop new construction projects. If yes, go to Step 16; otherwise go to Step 13.
16. Utilizing its knowledge, ES recommends candidate sites for new construction projects and the users finalize the type of new construction projects.
17. XPlanner estimates the facility requirements that can be satisfied by constructing new facilities and returns to Step 7.

As explained above, the problem-solving process of XPlanner involves an iterative process between the optimization model, the expert system and the user. The user may continue the consultation until an acceptable answer (facility management plan), which satisfies facility requirements generated by new missions, is found.

7. Possible Use of XPlanner by City Planners

XPlanner has been applied to a military setting, which is relatively well-defined area compared to cities. One can easily imagine the difficulties of developing this type of system for urban cities, where a variety of interest groups, many different activities, and facilities are involved with the problem-solving process. It would be worth applying this type of system, however, to more well-defined environments with somewhat clear objectives of problem-solving. A good candidate may include the spatial problem of managing public facilities such as recreational facilities.

The task of developing a facility management plan for recreational facilities is relatively well defined with a relatively limited number of players involved in decision making, planning goals, and decision constraints. Any cities should maintain a variety of recreational facilities, ranging from tennis courts and indoor gymnasiums to parks, within a relatively well-defined boundary.

City planners and administrators have the same concerns as the decision makers of Army installations: how to manage many different facilities in a given region while effectively supporting required recreational needs of citizens within a given budget and statutory requirements; which public facilities to build, close, renovate, and convert. They are concerned with current and future recreational trends and facility requirements of each type of recreational activities. The framework developed by XPlanner may be used to develop a comprehensive computer tool to support this problem.

8. Summary and Conclusions

As pointed out by O'Keefe (1985)[9], as operations research has shifted away from pure optimization models, it is likely that expert systems will shift away from pure symbolic processing systems and will increasingly employ optimization techniques. The coupling of ES and DSS basically takes two different forms: (1) integration of ES into the conventional DSS to provide qualitative reasoning capability and intelligent user interface and (2) integration of DS into the conventional ES to provide modeling capability. XPlanner takes the first type of coupling, in which ES helps the users input necessary parameters and interpret and modify the solutions by DSS. ES also provides judgmental capabilities to XPlanner to supplement qualitative tasks such as facility classification.

From the experience of designing XPlanner, it is believed that the intelligent interface provided by expert systems for the modeling tasks in DSS may stimulate planners to employ mathematical models more frequently and more easily in their problem-solving processes. As often pointed out by planners, modeling components tend to be treated as black boxes, inadequately recognizing the need for judgment by the users and concealing implicit judgments and assumptions from the users (Langendorf 1985)[5]. The integrated approach of XPlanner certainly provides a great improvement over the unaided use of modeling algorithms, encouraging easier use of quantitative modeling to support many planning decisions.

To develop XPlanner for field-level application, the hardware limitations should be resolved. XPlanner involves the complex numeric computations of integer programming in addition to the memory-consuming symbolic processing of the inference engine. The tasks involved in XPlanner require much more computer power than other types of traditional algorithmic processing tasks. XPlanner solves the hardware limitation to some degree by having the ES component reduce the size of the optimization problem.

Notes

1. In this example, the backward chaining process first fires Rule 2 because this rule contains the goal parameter *acceptability*. Then it checks if the condition part of Rule 2 is true, i.e. whether a zoning change is needed or not. This fires Rule 1 because this rule contains the parameter *zoning-change* and checks if the condition part of Rule 1 is true, i.e., if the land use is incompatible. Upon finding that the land use proposed by the development proposal is incompatible, it sets the value of the parameter *zoning-change* to *needed* and then sets the value of the goal parameter *acceptability* to *reject*.
2. In the ES shell Personal Consultant Plus, this can be accomplished by assigning ANTECEDENT property to necessary rules.
3. For instance, it is worthless for the model to consider the facilities classified by ES as disposable.

References

Richard Forsyth and Roy Rada. 1986. *Machine Learning Applications in Expert Systems and Information Retrieval*. Ellis Horwood Limited, West Sussex, England.

Sang-Yun Han and Tschangho John Kim. 1989. Can expert systems help with planning? *Journal* *of American Planning Association*, 55(3):296-308.

Greenberg Harold. 1971. *Integer Programming*. Academic Press. New York.

Donald W. Kroeber and Hugh J. Watson. 1987. *Computer-Based Information Systems*. Macmillan Publishing Company. New York.

Richard Langendorf. 1985. Computer and decision making. *Journal of American Planning Association*, 422-433. Autumn.

Ting-peng Liang. 1989. *A Composite Approach to Inducing Knowledge for Expert Systems Design*. Faculty Working Paper. Bureau of Economics and Business Research. University of Illinois at Urbana-Champaign. No. 89-1534.

Ting-peng Liang. 1988. Expert systems as decision aids: issues and strategies. *Journal of Informations Systems*, 2(2):41-50.

R. S. Michalski, J. H. Davis, V. S. Bisht, and J. B. Sinclair. 1985. Plant ds: an expert consulting system for the diagnosis of soybean diseases. In L. Steels and J. A. Campbell. editors. *Progress in Artificial Intelligence*. Ellis Horwood Limited. New York.

Robert M. O'keefe. 1985. Expert systems and operations research-Mutual benefits. *Journal of Operations Research Society*, 36(2):125-129.

Michael J. Shaw. 1988. *An Integrated Framework for Applying Machine Learning in Intelligent Decision Support Systems*. Faculty Working Paper. Bureau of Economics and Business Research. University of Illionis at Urbana-Champaign. No. 88-1485.

Herbert A. Simon. 1987. Two heads are better than one: The collaboration between AI and OR. *Interfaces*, 17(4):8-15.

R. E. Strauch. 1974. *A Critical Assessment of Quantitative Methodology as a Political Analysis Tool*. Land. Santa Monica.

George A. Tingley. 1987. Can MS/OR sell itself well enough? *Interfaces*, 17(4):41-52.

Efraim Turban and Paul R. Watkins. 1986. Integrating expert systems and decision support systems. *MIS Quarterly*, Jun.

John M. Wilson. 1985. Classification of models in operational research. *Journal of Operational Research Society*, 36(3):253-256.

Bayard E. Wynne. 1984. A domination sequance-MS/OR: DSS and the fifth generation. *Interfaces*, 14(3):51-58.