Primary vs. Secondary Decision Making: A Proposed Expansion of Model Management Systems Research

James R. Marsden, David E. Pingry, Robert D. St. Louis
University of Kentucky, University of Arizona, Arizona State University

ABSTRACT

A review of the historical development of model management systems leads us to suggest a needed expansion for future work. Our initial discussion of the evolution of model use in DSS is followed by a review of work to date on the development of model management systems. We argue that, in any but the most narrow DSS, model integration and model interactions necessitate determining the optimal scope of individual solution techniques and of the overall DSS. We identify several broad areas in model selection and integration which need to be addressed in future model management systems research, especially DSS scope determination. A detailed example is included to help illustrate the research issues and problems discussed in the earlier sections.

1. INTRODUCTION

Models are the basis for scientific decision making. Although literally hundreds of quantitative models have been developed, their application remains limited. In fact, the simple spreadsheet model is perhaps the only model which has truly achieved widespread application. The eagerness with which managers have embraced spreadsheets suggests that there is a desire to use models, but that the more sophisticated models are still too difficult to use or are not seen as justifying their cost.

This paper examines the use of quantitative models in decision support systems (DSS) and suggests an extension of model management systems (MMS) research. A critical argument we will make later is that this redirection is a necessary condition if quantitative models are to ever achieve widespread use. The next section focuses on the evolution of the use of models in DSS. Section 3 provides a review of relevant work to date on developing model management systems. Section 4 provides the key elements of our argument, including the specification of problems in model selection and integration which have not yet been dealt with in the MMS literature. This section also suggests an approach for dealing with these problems, including detailed consideration of the critical issues of scope. Section 5 provides an illustrative example to help clarify the issues raised and problems discussed here. The example also provides an opportunity to emphasize our suggested methodologies for research and implementation. The concluding section of the paper proposes that MMS research be expanded to include model interaction and DSS scope determination issues.

2. THE EVOLUTION OF THE MODEL IN DECISION SUPPORT SYSTEMS

Elam et al [11] suggested that significant benefits can be realized by developing more generalized decision aids that access, utilize, adapt, and integrate models. The benefits cited included: (1) easier model development as a result of consistent definition and representation of models within an organization; (2) better utilization and diffusing of modeling knowledge; and (3) the establishment of a concrete basis for estimating development costs, assigning priorities to subtasks, and selection personnel.

Sprague and Carlson [32] introduced a useful framework for DSS research, comparing the perspectives of the end-user, the builder, and the toolsmith. From the viewpoint of the model builder, the key capabilities of a model subsystem are that it can: (1) create models easily; (2) catalog and maintain a wide range of models, supporting all levels of management; (3) interrelate models with appropriate linkages through a database; (4) integrate and access model building blocks; and (5) manage a model as a database.

Boczek, Holapple, and Whinston [3], envisioning how the advanced capability of artificial intelligence could be applied to DSS, proposed a "generic DSS" composed of three sub-systems: (1) a language system which is the sum of all linguistic facilities; (2) a knowledge system which is the body of knowledge about a problem domain; and (3) a problem processing system (PPS) which is the mediating mechanism between expressions of knowledge in the knowledge system and expressions of problems in the language sys-
Problem recognition and model formulation are done in the PPS.

Bosmans and Sol [5] proposed adding a decision process to the problem processing system in Bonczek, Holisapple, and Whinston's framework. Bosmans and Sol's PPS consisted of four components: problem, conceptual model empirical model, and solution. These four components correspond to the four components of the typical decision process (problem recognition, conceptualization, specification, and solution procurement).

After observing the different generations of DSS, Liang [24] noted that there has been a trend toward collecting models to form a model base management system (MBMS). According to Liang, the advantages of a MBMS are: (1) redundancy can be removed; (2) models can be shared; and (3) flexibility can be increased.

3. DEVELOPMENT OF MODEL MANAGEMENT SYSTEMS (MMS) AND THE BEGINNING OF CONCERN ABOUT META-DECISION PROBLEMS

Early work in MMS included the efforts of Will [36] and Sprague and Carlson [32] with strong emphasis placed on the functionality of the MMS approach.

Recent efforts include the contributions of Dolk and Konasymaki [9], Elam and Konasymaki [12], Blanning [1], Geoffrion [15], and Dutta and Basu [10]. Dolk and Konasymaki [9] proposed a model abstraction scheme for model representation. Resembling the frame concept in artificial intelligence, their model abstraction consists of three components: (1) the data objects - an enumeration of the data items and types comprising the model; (2) the procedures - a list of each procedure along with the data objects accessed by the procedure and the data objects returned by the procedure; and (3) the assertions - propositions or conjectures about data objects, procedures, and possibly their various interrelationships. Their work treats data types, data items, and procedures as predicates while assertions are well-formed formulas in the predicate calculus. In fact, using this model abstraction framework, Klein, Konasymaki, and Beck [19] proposed a linear goal programming method to select models. One key difficulty they point out, however, is that of coming up with attributes to clearly differentiate models.

Using relational database concepts, Blanning [1] treated a model as a properly restricted subset of the Cartesian cross product of the model inputs and its outputs. The model bank may then be viewed as a set of virtual relations with input/output attributes and functional relationships between them.

Elam and Konasymaki [12] adopted the entity relationship approach. The MMS they proposed has five components: analyzer, builder, interrogator, processor, and knowledge base. The model is treated as a subroutine and the goal is to build the capability of automatic modeling into the knowledge base using a graphical language called SI-NET (Structures Inheritance Network). Geoffrion [15] proposed the methodology of structured modeling which gives definitions to all of the elements comprising a model. The definitions have the following special properties: (1) they are typed; (2) they are correlated - the interdependencies are explicit; (3) specified types are value-bearing. The definitions are grouped by definitional similarity with the resulting group organized hierarchically by conceptual similarity. Checks are also made to insure that the overall system is free of circularity.

Dutta and Basu [10] developed a formal representation of models using a logic framework and provided methods for machine representation of computational models. Liang [24] brought the users' roles into the MMS framework. He portrayed an MMS as having three levels: external schema, logical configuration, and physical configuration. These correspond to the three different user roles from Sprague and Carlson: information user, model builder, and toolmith. The external schema and logical configuration are linked through a help module where the model builder enters explanatory notes and the information user receives the characteristics of the model whenever a choice is made.

As suggested by this brief literature review, MMS have to date denoted generalized software systems that offer a wide range of models and allow for flexible access, update, and change of the model base. Yet little attention has been given to certain aspects of decision support, especially the critical, and logically prior problems, of determining the exact models to include in a model base. Bosman [4] suggested that in order to increase the effectiveness of DSS model components as cognitive aids, we should: (1) get rid of the idea that there is only one model that describes all phases of the decision process; (2) stress the importance of informal procedures (meta-decision) by constructing descriptive models using rule-based systems or so called "process models;" and (3) construct models for the design phases of the decision process.

Wedley and Field [33] argued that the decisions managers make before they begin to solve a problem often severely constrain the set of feasible solutions. Hence, these prior decisions can be crucial to the success of subsequent courses of action chosen using a DSS. They present an example of a pre-decision support system for helping managers to select a decision style, method, and membership to be used in solving a problem." Kotteman [21] decomposed managerial problem solving into three basic types of tasks: 1) problem opportunity recognition, 2) meta-decision making, and 3) primary decision making. Noting that current DSS research and development have almost exclusively addressed primary decision making, he argued that "DSS and AI offer techniques with which to develop systems
in support of meta-decision making and problem recognition." Through the issues they raise, Wedley and Field and Kotteman set the stage for our concern—the decision regarding what solution structures to include in the model base. In the next section we begin our formal consideration of this meta-decision making problem.

4. CHOOSING AND INTEGRATING SOLUTION STRUCTURES IN THE MODEL BASE—A META-DECISION PROBLEM

The suggestions of Kotteman [21] lead us to the view that our problem can be modelled as a set of production rules. In this context, the inference mechanism of the expert system is simply a contingency model analyzer. The decision making situation, as defined by a set of situational variables, is mapped into a decision making method defined by a set of decision techniques and decision aids. The left hand sides of the production rules are values of situational variables, and the right hand sides are names of corresponding techniques and/or aids. As discussed in greater detail later, the use of this type of approach proves beneficial for clearly explaining the reasoning path(s) for choosing models to include in the model base.

To aid clarity, we begin by setting out the following definitions of terms to be used throughout this discussion:

SOLUTION STRUCTURE—a specific problem solving technique such as "linear programming simplex method," "assignment problem algorithm," "least squares linear regression estimation," or "one way analysis of variance (ANOVA);"

PROBLEM STRUCTURE—specific problem description such as "estimating inventory with given historically observed values," "assigning m employees with known characteristics to m machines with given characteristics;"

SOLUTION STRUCTURE SCOPE—the set of problem structures to which a specified solution structure is to be applied in the specific situation under consideration;

DSS SCOPE—the set of problem structures to be addressed by any of the solution structures in the DSS.

Two basic arguments run throughout the discussion that follows:

1) optimal model selection is crucial if we are to take full advantage of recent innovations in model management techniques;

2) optimal model selection is equivalent to determining the optimal scope for solution structures in decision support systems.

We espouse a more effective use of existing technologies. Previous researchers in model management have developed procedures for matching data to techniques (e.g., Kumar and Hsu [23] for selecting forecasting procedures, Murphy and Stohr [29] on linear program model construction and Bu-Hulaiga and Jain on model selection and sequencing) and for "matching" assumptions to data (that is, checking whether or not the data indicate specific assumptions are rejected for the given data set, Gale [13]). Our work focuses on the problem of determining precisely what models belong in the model base. Doing so requires situational specific data and information to perform the necessary analyses. Rather than focusing on selecting one solution technique from a specified set of solution techniques, we focus on the prior issue of how to optimally specify the set of solution techniques. Given an informational data set on situational variables, we attempt to determine the solution structure space and the scope of each included solution structure. This is opposed to taking the solution structure space as given and then searching the given space for the best or most appropriate solution structure for a given problem. As noted earlier, our procedure involves determining the optimal mappings from left hand sides of production rules (situational variables) to right hand sides of production rules (solution structures). In doing so, we are determining the scope of the knowledge systems or DSS. As suggested earlier, the scopes we refer to are situational specific. We are not considering what problems a given solution procedure can be validly applied to, but rather, in a specified situation, what problems (if any) should be tackled with a specific solution structure. Since the costs and benefits from problem solution are situational specific, the determination of solution structure scope and DSS scope are situation specific.

The generic term "problem solution" provides little information of value. The interesting questions involve determining the cost of solving the problem and the value or return achieved from solving it. For most problems there are a variety of possible solution techniques yielding anything from poor approximations to the true optimal solution. Associated with each of this variety of solution procedures is a solution cost. Though we might tend to associate more accurate solutions with higher solution costs (e.g., more careful or complete search, more time consuming analysis) such a relationship is by no means a certainty. Relatively cheap solution procedures can often yield better solutions than can more costly solution procedures.

Suppose, for example, that we consider a decision maker facing five distinct problem structures, P1, P2, P3, P4, and P5. In typical business applications, the set of problem structures might include inventory control, employee assignment, warehouse delivery, production input demand, and sales forecasting. The decision maker might be aware of and have potential access to the following solution procedures: economic order quantity (EOQ), assignment algorithm (AA), transportation algorithm (TA), line-
ar programming simplex algorithm (LP), linear regression statistical estimation package (REGRESS), and non-linear regression statistical estimation procedures (NONLIN). The following non-exhaustive problem structure-solution procedure combinations illustrate four possible mixes:

(i) P1 - EQQ
(ii) P1 - EQQ
(iii) P1 - EQQ
(iv) P1 - EQQ
P2 - AA
P2 - AA
P2, P3, P4 - LF
P2, P3, P4 - LF
P5 - REGRESS
P5 - NONLIN
P5 - REGRESS
P5 - NONLIN

Under grouping three, EQQ would be used to solve the inventory problem, LP to solve the assignment, transportation, and input demand determination problems, and REGRESS to solve the forecasting problem. For the generated solutions to be optimal for a specific setting, a series of specific conditions must be satisfied. These conditions include: 1) the true sales model is linear in transformations of inputs into outputs, and 2) the production process model is linear in transformations of inputs into outputs.

Though the individual solution structure scopes would vary across the five cases, the DSS scope would not. This is so since each DSS is intended to deal with identical problem structure sets. If one or more problem structures were deleted (or additional ones added) for solution by a specified DSS, then the DSS scopes would vary.

A DSS is, in fact, an organized set of solution structures that can be consulted by a decision maker to help deal with a problem. A DSS may consist of a single solution procedure or a portfolio of such procedures. Thus, the solution structures contained in (i) through (iv) above represent four differing DSS portfolios. The first, (i), is composed of EQQ, AA, TA, LP, REGRESS while the third is composed of a subset of (i) including only EQQ, LP, and REGRESS.

The decision maker actually faces a three step procedure to deal with anticipated problem occurrences. First, prior to problem occurrence, he or she must decide what techniques to include in the DSS portfolio. Second, when a problem arises, the decision maker must choose which, if any, of the solution structures to use in attempting to solve the problem. The third step involves the determination of how to utilize the output generated from the DSS in dealing with the problem. The first two steps are our focus here. The first involves the determination of the scope of each of the solution techniques and the scope of the DSS portfolio chosen. The second involves the assignment of specific solution procedures to particular problem occurrences. Each of these tasks involves repetitive applications of expert judgment that are logical candidates for rule-based expert system modelling as suggested in our discussion above. For example, in the second step just described, the "IF" condition of the rules would involve specific requirements of the quantitative techniques included in the DSS portfolio. Examples might include the following IF conditions:

IF (objective function is linear) THEN (...) IF (regression forecasting equation is non-linear) THEN (...)

What we are suggesting is a two stage ES/DSS application. In the first stage, an ES is used to formulate the solution structures to be included in the DSS portfolio. Using information on occurrences of specific problem structures, costs associated with such occurrences, available solution structures, and costs of obtaining and utilizing such solution structures, the first step ES is used to determine the form and content of the DSS portfolio (for a more complete discussion of such an approach see Marsden and Pingry [27]). In the second (and continuing stage), the ES is used to determine the scope of the DSS. When a particular problem occurs, information is entered into an ES which is used to determine what, if any, solution structure in the DSS is appropriate for use in solving the problem. For each problem occurrence, it must be determined whether to ignore the problem or how to seek a solution. The first option would be chosen when it is anticipated that solving the problem costs more than solving the problem is worth. When the second option is selected, the determination must also be made as to which solution structure(s) to utilize in seeking problem solution(s). This determination necessitates the consideration of standard condition or assumption validation prior to choice of solution method. Further, the need will commonly arise for conflict resolution strategies to deal with situations where the conditions of more than one solution structure are satisfied by the given problem structure. Referring back to our earlier example, consider the DSS noted as (ii) consisting of EQQ, AA, TA, LP and NONLIN. If a transportation problem occurred, its structure would satisfy the conditions of both the TA and LP solution structures since the former is a subset of the latter. In most cases, there would be no gains from utilizing the LP rather than the TA solution procedure. An example of where the LP might be preferred is where the decision maker has familiarity with problem formulation and result analysis of an LP (including sensitivity analysis) but no familiarity with use of the TA. If training costs outweigh gains from using the more specialized and faster TA solution procedure, the LP framework would be preferred.
The formulation of the two step ES process—the first step focusing on DSS portfolio selection and the second step focusing on determining the solution structure(s) for given problem occurrences—requires that the issues of economies of scale and of scope be directly dealt with. The former are savings in the average cost of problem solution (e.g., fixed cost spread over more occurrences) achieved by applying a solution structure to a larger number of occurrences of a given problem structure. The latter are cost savings achieved by applying a given solution structure to an increasing number of problem structures. By solution structure scope we mean the set of problems to which a specified solution procedure is to be applied. Constructing the first stage ES would be based on information relating to the selected scope of each solution procedure and the expected number of problem occurrences within the scope of each procedure. The second stage ES would involve procedures to determine whether or not a specified problem occurrence is within any of the scopes specified for the solution procedures included in the DSS constructed using the first stage ES. As suggested earlier, determining the scopes for each solution procedure included in the DSS would require conflict resolution procedures. That is, when a problem occurs which potentially could be in the scope of more than one solution procedure, the scope specification determined by the first stage ES would normally assign the occurrence to only one solution procedure. Assignment to more than one solution procedure (that is, overlapping solution procedure scopes) could occur if there were net gains to be achieved through information provided to the decision maker on multiple solution technique outcomes.

Determining the portfolio of solution structures and their individual problem scopes is the DSS design problem. Using an ES to perform these courses requires the specific criteria in the form of the knowledge base, rules, and the inference engine incorporated in the expert system. The criteria to be represented include the variations and conditions inherent in the DSS portfolio of solution procedures. As noted earlier, the LP procedure requires a linear objective function and linear constraint inequalities. The second stage ES would require a knowledge base, rules, and inference engine capable of determining whether the specified problem is in one or more of the solution scopes. If the ES determines that the problem is not in the scope of any solution procedure, then the DSS would not be applicable. Otherwise the ES would specify which solution procedure(s) should be applied to the specified problem.

Accomplishing these goals requires detailed analysis both historical data on the precise nature of problem structures to potentially be addressed by a DSS and its included solution structures. Though it is no doubt fruitless to attempt precise analytic steps to follow in such an analysis, we can outline general procedures. In the next section we use an example from the Arizona Unemployment Insurance Agency (UI) to illustrate the processes we have been discussing.

5. FORECASTING EXAMPLE

This section uses a forecasting example to illustrate our proposal. The example is based on the authors' work with the Arizona Unemployment Insurance (UI) Agency, which collects taxes from employers and pays benefits to eligible unemployed workers (claimants). The funds which the UI Agency receives for administering the program come from the U.S. Department of Labor, and depend upon the workload of the agency (the number of employers taxed, the number of checks written, the eligibility issues adjudicated, etc.). The dollars paid to claimants come from taxes deposited into the Arizona UI Trust Fund, and depend upon the number of claims paid and the amount of each claim. In order to manage the program, the UI Agency must forecast each of the following: (1) the total amount of dollars that will be paid to claimants, (2) the total amount of dollars that will be collected from employers and deposited into the trust fund, and (3) the various workload items for which the agency receives administrative funding from the Department of Labor.

The forecast for dollars paid to claimants is extremely important to the UI administration. The funds paid to claimants come from taxes collected from employers through the UI tax and placed in the trust fund. The tax rate can be changed only on January 1 of each year, and employers must be notified of the new rate by November 1 of the prior year. Moreover, revenues from the new tax rates are not collected until April 30 of the new tax year. Hence the administration cannot change the impact of a tax rate until 18 months after the rate is set.

The workload forecast also is very important to the UI administration. The funding which the administration receives for administering the program is determined by such workload items as number of checks written, number of eligibility issues adjudicated, and number of employers doing business in the state. As would be expected, the agency's workload tends to vary seasonally and cyclically. As a consequence, the agency's staff consists of both part-time and full-time workers. Part-time workers understand that they are going to be used only to meet peak demands. Full-time workers expect to have a continuing, year-round job. In order to know how many full-time and part-time workers to hire, and in order not to exceed its budget, the agency must have a workload forecast.

This situation is such that scope of the DSS is easily determined since the cost of each problem is prohibitive unless techniques are provided to deal with each one. In order to staff appropriately, the agency must be able to forecast work-
load, both for the short and the long term. In order to appropriately manage the trust fund, the agency must be able to forecast the amount of benefits that will be paid and the amount of taxes that will be collected. And finally, in order to adequately respond to legislative inquiries, the agency must be able to simulate the impacts of programmatic changes such as increasing the weekly benefit amount by $20, lengthening the disqualification period for voluntarily leaving a job, increasing the number of job contacts that a claimant must make, etc.

Determining the optimal or even an acceptable domain for the model management system within the DSS is a much more difficult problem. Various authors have referred to this type of decision as a meta-decision [Mintzberg et al, 1976], a secondary decision [White, 1975] and a predis- cision [Wedley and Field, 1984], and have suggested that a DSS should be able to help managers confront this type of situation.

Hershauer and Stuart [17] specifically examine predisdictions relating to forecasting and conclude that:

"the key decisions are the metadecisions. These are the decisions which determine how to approach the focus decisions and how to structure models in the computer system. Some operations management executives have abdicated these meta-decisions to computer specialists or consultants or the system. This is unnecessary, dangerous, and dysfunctional for the organization."

Unfortunately, Hershauer and Stewart [17] do not offer any guidance as to how a DSS might help managers make this type of predetermination.

Here we look at how to formalize the knowledge required to develop a meta-DSS, and pay particular attention to the implications for model integration. In our example two different groups as well as two different models are involved. Workload forecasts are used primarily by the resources management group. Trust fund forecasts and simulations are used primarily by the actuarial group. In order to decide what models ought to be included in the model base, it is important to examine general forecasting issues, agency specific situational factors, and the interactions between the actuarial and resource management groups.

In the specific situation being discussed here, there are also important issues, including interaction effects that would have to be dealt with if the optimal model set were to be chosen for the model base. For example, there is a need for both short and long run forecasting with the workload model utilizing inputs from the trust fund model for average duration, total weeks claimed, unemployment rates, before translating them into initial claims and other work- load variables. While workload variables are needed on a quarterly basis, trust fund variables are important on an annual basis. The two, however, must be consistent in corresponding annual series. Further, since the workload model deals with variables that are functions of pay- ments (and both are functions of the number of covered employees), the workload and payout mod- els are highly interactive.

The repetitive nature of the problems faced by UI are such that we would also expect economies of scale to occur. Further, since forecasting lies at the heart of several problems, economies of scope may also play an important role in the determination of an optimal model base.

In general, forecasting approaches can be catego- rized as qualitative/judgmental, time series/ extrapolative, or causal. Each of these tech- niques has different data, computer support, and personnel support characteristics. They also differ in their accuracy, their ability to handle "what if" situations, the difficulty of inte- grating different models, and their orienta- tion toward top down or bottom up forecasts. Each of these factors, as well as the specific organizational context must be considered when deciding whether a particular model should be included in the model base for a DSS.

Georgoff and Murdick [16] present a managers' guide to forecasting. This guide was used by both Nute, Mann and Brewer [30] and Kumar and Hsu [23] to construct a knowledge base for select- ing a forecasting system. If all feasible forecasting techniques and their properties are included in such a table, a similar procedure could be used to identify which models are can- didates for inclusion in a model base. That is, an ES or DSS could be used to select a set of potential or feasible techniques for each prob- lem in the scope of the DSS.

The advantages of using a DSS at this stage, as pointed out by Kumar and Hsu, are: 1) a more complete consideration of alternative tech- niques; 2) suggestions for combining appropri- ate complementary techniques; 3) the ability to apply heuristics; 4) the ability to show the path of reasoning; and 5) freedom from the selec- tion biases which managers frequently devel- op. That is, the DSS should allow managers to quickly identify appropriate forecasting tech- niques to help with their problems. The model base also may be different for each organiza- tion, since the importance of the criteria associated with the forecasting technique may vary for each organization.

In our example, the result of the first applica- tion of the DSS might be the selection of Box- Jenkins methods, regression methods, econometric methods, time series decomposition methods, leading indicators, scenario methods, and gener- al purpose simulators. If resources were unlimited, all of these techniques probably would be included in the model base. The issue then be- comes whether it is worth while to use any or all of these methods. Though not appropriate for the UI example, in many instances the best solution might be simply to ignore or not solve
the problem. Thus far, model management system research has not addressed this issue.

To answer this more difficult and relevant question we must know: The frequency of each problem occurrence, the cost associated with each problem occurrence, the scope of each solution structure, the cost of using each solution structure, and the nature of the interactions between solution structures. Moreover, because the cost of using each technique may be dependent on being able to integrate models, it is important to determine the difficulty of: 1) automatically developing a conceptual schema for combined/integrated models; and 2) solving the combined model given only the information on the individual models.

The cost implications of model integration are especially important when determining which models ought to be included in the model base. Proposals for facilitating model integration for an extant DSS have been proposed by several authors (e.g., Kottmann and Dolk [22] and Bradley and Clemence [6]), but no results have as yet been published on the cost implications of being able to effectively integrate models.

The task of determining which models are cost effective obviously is as difficult as the task of determining which techniques are appropriate for the model base. An approach for dealing with this problem can be found in Marsden, Pingry and St. Louis [28]. That issue is not addressed in this paper. Instead, the purpose of this example was simply to show how a DSS could help managers select a set of appropriate techniques to consider for inclusion in the model base.

SUMMARY AND CONCLUSIONS

A number of authors have suggested that a DSS should be able to help managers with both meta-decisions and focused-decisions. Many papers have been written on how a DSS could be used to help managers make the focused-decision of selecting a particular model to help solve a specific problem, but none have been written on how a DSS might help a manager make the meta-decision which models ought to be included in the model base. This paper shows how a DSS could be used to help managers make that meta-decision, and shows that the same technologies can be used to address both the meta and focused decisions.

In order to determine what model ought to be included in the model base, the decision must be viewed as a two step process: 1) determine which models are potentially useful for solving the problems in the scope of the desired DSS; and 2) determine which models are cost-effective to include in the scope of the DSS. After the meta-decision is made, attention can be shifted to the focus-decision of selecting a specific model for a specific problem.

The paper also shows that a key to determining which models should be included in the domain of the DSS is the consideration of interactions in the scope of the DSS. This is the counterpart to the issue of model integration which has been addressed in the model management systems literature.

Based on these results, it appears that model management system research ought to be broadened to include meta-decisions, and that decision support systems might prove even more helpful for meta-decisions than focused decisions.

REFERENCES


[33] Wedley, W. C. and R. H. G. Field, "A Pre-
decision Support System," Academy of Man-

[34] Will, H., "Model Management Systems," in
Information Systems and Organizational
Structure, eds. E. Rochla and N. Szypenski
(Berlin: W. DeGraeter), 1975.