

Model Management for Group Decision Support

By: Ting-Peng Liang
 Department of Accountancy
 University of Illinois at
 Urbana-Champaign
 1206 South Sixth Street
 Champaign, IL 61820

Abstract

Since models play a critical role in human decision processes, model management is considered a very important function for decision support. This article examines how model management systems can be designed to support group problem-solving. First, basic concepts of model management and functional requirements for group model management systems are described. Then, an architecture for group model management systems design is presented. Finally, major implementation issues are discussed.

Keywords: Model management systems, group decision support, decision support systems

ACM Categories: C.2.4, H.1.0, H.2.4, H.4.2, I.6.0

Introduction

Due to the nature of human society, group problem-solving plays an important role in modern organizations. Recently, successful applications of individual decision support systems (DSS) have raised much interest in developing group decision support systems (GDSS). The purpose of these systems is to combine information and decision technologies to facilitate solutions of semi-structured or unstructured problems by a group of decision makers working together. Various issues have been studied. For example, Huber (1984) describes GDSS capabilities, delivery modes, and various design strategies. DeSanctis and Gallupe (1985; 1987) present four basic components of GDSS, design and im-

plementation issues, and several research directions. Bui and Jarke (1986) discuss the communication requirements. Chang (1986), Decker (1987), and Rathwell and Burns (1985) investigate distributed problem-solving techniques. Vogel, et al. (1987) examine determinants of success based on the results obtained from the Decision and Planning Laboratory at the University of Arizona. Most of these researchers suggest that the process and performance of group decision making can be improved by GDSS. Empirical studies also indicate that GDSSs support idea generation in brainstorming processes and enhance the quality of group decision making. For instance, Nunamaker, et al. (1987) report high levels of satisfaction with electronic brainstorming as a tool for generating ideas during organizational planning sessions. Gallupe, et al. (1988) show enhanced decision quality when GDSSs are provided.

Although these previous works have portrayed a general framework for GDSS, one key issue that remains unclear is how model management capabilities can be integrated into a GDSS. Model management is an important function of DSS. Because of human cognitive limitations, people usually use models to help them understand, organize, study, and solve problems (Simon, 1981). This is particularly true when the problem to be solved is complex and difficult. In this case, computer-based decision models may be crucial to the quality of the decision. Most GDSSs presented so far tend to serve as a communication blackboard on which ideas can be generated, information can be shared, and consensus may be reached by using group techniques such as voting and preference ranking. This type of system certainly can provide valuable support to group meetings. In some situations, however, a higher level of modeling support may be desirable, especially when there are conflicts to be resolved (DeSanctis and Gallupe, 1987; Goncalves, 1985; Gray, 1987; Huber, 1984; Jarke, 1986).

For example, when a group of managers fails to reach an agreement on the forecasted sales for next year, it is usually not a matter of voting or preference ranking. Nor will a multi-attributed decision model or a game-theoretical approach be appropriate for resolving conflict. In fact, the disagreement may result from differences in model assumptions or in the selection of models. Therefore, it would be very useful if the GDSS provided functions that allowed the managers to examine what models were used to generate

their sales figures, what assumptions were behind these models, and how these models were evaluated; then, if necessary, it would help them develop a new sales forecasting model acceptable to all group members. In other words, the system needs to facilitate the group in *examining, manipulating, and developing* decision models.

The remainder of this article investigates how model management functions can be provided to support group decision making. First, basic model management concepts will be described. Then, model management requirements for GDSS will be presented. Finally, design of group model management systems (GMMS) and the implementation issues will be discussed.

Model Management Concepts

A model is an abstraction of a specific problem or a class of problems. Most models designed to support today's human decision making are complicated, knowledge-intensive, and computer-based. Because of the important role models play in human decision processes, literature in DSS usually considers models as a valuable organizational resource that should be managed properly. One way to manage models in DSS is to incorporate a model management subsystem, which includes a model base and model base management system (MBMS). This approach has been well-accepted by DSS researchers (e.g., Bonczek, et al., 1980; Dolk and Konsynski, 1984; Elam and Konsynski, 1982; 1986; 1987; Sprague and Carlson, 1982; Stohr and Tanniru, 1980).

A model base is a collection of computer-based decision models. Its function is similar to a database, except that the stored objects are models. In general, a model base is both integrated and shared. By "integrated" we mean that the model base may be considered as a unification of many otherwise distinct models with redundancy among them partially or wholly eliminated. By "shared" we mean that any model in the model base may be used for formulating another model or be accessed by any authorized users. An integrated and shared model base can provide at least the following advantages (Liang, 1985):

1. **Redundancy will be reduced.** Since models are shared, redundant storage will be identified and removed.

2. **Consistency will be increased.** More than one decision maker may share the same model and, hence, reduce the likelihood of using inconsistent information generated from different models or different versions of a model.
3. **Flexibility will be increased.** When a model needs to be upgraded, it needs to be done only once. In addition, any upgrade or change in the model base will be made available to all users automatically.
4. **Control over the decision process will be improved.** Since models play an important role in human decision processes, control over the decision process may be improved by controlling the quality of the models adopted.

A model management system (MMS) is a software system that handles all access to the model base. For an individual DSS, an MMS provides at least five basic functions (Sprague and Watson, 1975; Will, 1975):

1. **Construction of new models** — It provides an environment in which new models can be developed with minimum effort.
2. **Storage of existing models** — It maintains a model base in which decision models are stored.
3. **Access and retrieval of existing models** — It facilitates the utilization of decision models in the model base.
4. **Execution of existing models** — It executes an existing model and reports outputs of the model.
5. **Maintenance of existing models** — It supports the update and modification of existing models.

In addition to these basic functions, recent research in MMS has emphasized advanced features including model integration and model selection. Mechanisms for model integration help decision makers develop new models by combining existing ones in a proper sequence. In other words, existing models in the model base are considered building blocks upon which larger models can be constructed. This capability is very useful when ad hoc models are desired. Mechanisms for model selection help decision makers choose appropriate models for model in-

tegration and problem-solving. Detailed discussion on these issues can be found in Elam and Konsynski (1987), Geoffrion (1987), Klein (1986), Liang (1986; 1988a; 1988b), and Liang and Jones (1988).

Model Management Requirements for Group Decision Support

Since group and individual decision processes are different in many aspects, functional requirements of a GDSS may be different from those of an individual DSS. For example, a typical group decision process usually involves stages, such as orientation, conflict, negotiation, emergence, and reinforcement, not shown in individual decision making (e.g., Fisher, 1974; Goncalves, 1985). In order to support these activities, GDSSs need to provide additional functions not available in individual DSSs, including communication, information sharing, and conflict resolution. These additional functions result in extra model management requirements, which can be described from three aspects — *scope of support, level of support, and decision environment*.

Scope of support

The scope of support defines the circumstances in which GMMS support is desirable. It can be portrayed from two dimensions: a *structured or unstructured task* and a *data-oriented or model-oriented process*.

In an individual DSS, quantitative models are frequently used to deal with the well-structured portion of a decision process. The unstructured portion is handled by the decision maker who defines the goal and makes the final decision. In group processes, however, a group of decision makers needs to reach an agreement since each of them may have different goals and different perspectives on the role of modeling. Hence, many political issues may be involved (e.g., Dutton and Kraemer, 1983), and model management support needs to be expanded to incorporate some unstructured tasks in the process. Negotiation and conflict resolution, for instance, are among the major unstructured tasks that need model management support.

In addition to the structured/unstructured dimension, group decision processes may also be classified into *data-oriented* and *model-oriented*. A

data-oriented process involves heavy communication of data and relatively light use of quantitative models. A typical example is brainstorming for new ideas. The GDSS keeps track of the ideas generated and facilitates the discussion process but may provide few models for idea evaluation. A model-oriented process, on the other hand, involves steps for formulating quantitative models or requiring extensive use of these models.

For example, when the accounting, production, and marketing managers in a company are working together to develop a strategic business plan based on the anticipated demand and available resources, this process usually is model-oriented. As shown in Figure 1, product pricing, sales forecasting, and market segmentation and penetration models may be used by the marketing managers. From the accounting perspective, however, pricing decisions may need to consider costs determined by a cost allocation model. In addition, any market segmentation or product differentiation strategies that require capital investment would need to go through a capital budgeting process. To production managers, the cost allocation and capital budgeting information must be integrated into the capacity planning, production scheduling, and inventory control models. The marketing strategy should also be grounded on production capacity. It is obvious that in this process different parties representing different interests, having different factual evidences and value judgement, and *armed with different decision models* need to communicate and negotiate to reach an agreement.

During the decision process, the managers not only use models to facilitate their analyses but also must convince other members that the results generated by the model are of value. The accounting and production managers may question the forecasted sales estimated by the marketing department. The marketing managers may want to know how the production process is scheduled and what criteria are used in the capital budgeting process. If a particular model is considered inappropriate, then the group may want to modify it or choose another one.

The taxonomy indicates at least two different ways to provide group model management support. First, in addition to individual decision models, the system can provide group models such as various voting and structured group decision techniques. These models can be used

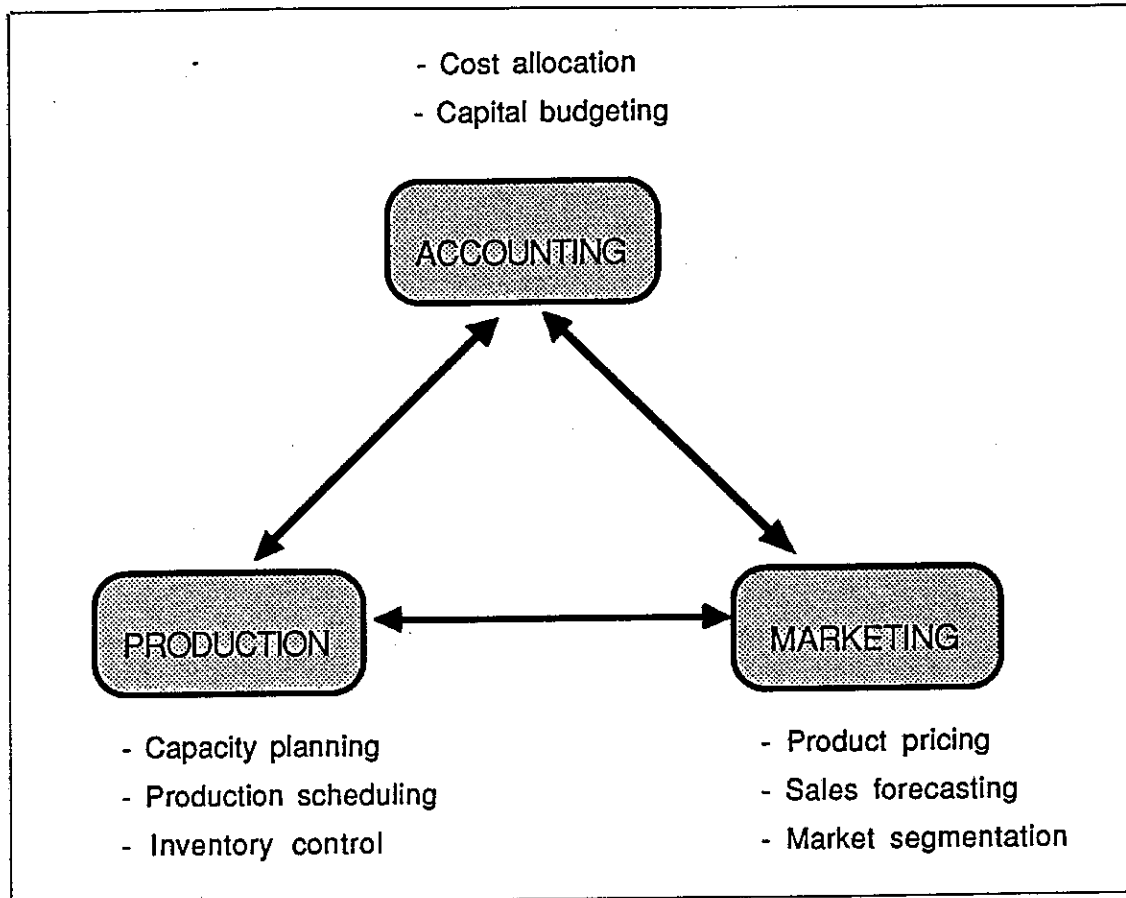


Figure 1. A Model-Oriented Group Decision

to facilitate data-oriented processes in information gathering, idea structuring and analysis, and choice (Applegate, et al., 1987). Second, the system can provide group model management functions that enable model development and use by groups in model-oriented processes. Depending on the complexity and the modeling requirement of the decision process, these functions may vary from simply sharing model inputs and outputs to a joint construction of models that integrate various components developed by members. In other words, different levels of support may be desired in different situations.

Level of support

Several factors may affect the functional requirements of GMMS. For example, in the case where group members are already familiar with each other's interests and decision models and the degree of goal conflict is low, then functions that allow them to share the model inputs and

outputs and to exchange comments may be adequate. Otherwise, more powerful functions would be necessary for them to examine the assumptions of models, explore model structures, and integrate relevant models for a joint construction of new models. In a recent article, DeSanctis and Gallupe (1987) also state that for different tasks three different levels of GDSS are desired:

1. Level-1 GDSSs provide technical features aimed at removing common communication barriers, such as large screens for instantaneous display of ideas, voting specification and compilation, anonymous input of ideas, and electronic message exchange between members.
2. Level-2 GDSSs provide decision modeling and group decision techniques aimed at reducing uncertainty and "noise" that occur in the group decision process.

3. Level-3 GDSSs are characterized by machine-induced group communication patterns and can include expert advice in the selection and arrangement of rules to be applied during a meeting.

In line with this taxonomy, the functional support of GMMS can be classified into three levels: *communication, modeling and negotiation*, and *intelligent support*. Table 1 outlines the desired capabilities at each level.

In group problem-solving, there are two generic bases for disagreement: uncertainty due to incomplete information and conflict objectives (Goncalves, 1985). A first-level GMMS focuses on facilitating communications to simplify model utilization and information sharing in group meetings, with a goal of sharing and using numeric, textual, and relational information to reduce the disagreement caused by incomplete information. With this level of support, models can be executed, and the input and output information of the models can be disseminated to interested members. But the models themselves are not shared. In other words, each member is allowed to access only his or her own models. Model management capabilities required at this level include execution of individual models, information sharing mechanisms, and model access and retrieval control. Selected group voting and preference ranking models may also be included in the group model base. Given the previous business planning example, a first-level GMMS will support marketing managers' use of product pricing, sales forecasting, and market segmentation models and the dissemination of their results to accounting and production managers. They may

not be allowed to use or modify accounting and production models.

The second-level GMMS focuses on modeling and how models can be shared to facilitate negotiation when conflict exists. The shared decision models serve as a basis in which group members examine various scenarios, conduct what-if analysis, debate assumptions of the models, and modify existing models or create new models to resolve conflict.

In addition to the first-level functions, a second-level GMMS needs to provide several capabilities. First, the system supports model sharing capabilities. Each group member is allowed to examine and use the models created by other members.

Second, the system supports the integration of data and models owned by the same or different members. Because model execution usually needs data stored in the database, the data-model integration should enhance the process of modeling and model utilization by simplifying the linkage between the two.

Third, the system supports model integration. Model integration is a specific form of model construction. Instead of building new models from scratch, it allows group members to formulate a composite model by organizing existing component models, each of which is capable of solving part of the problem. The component models to be integrated in GMMS may be developed and owned by an individual member or several different members.

Fourth, the system supports modification (or maintenance) of existing models and creation of new

Table 1. Three Levels of Model Management Support

Level	Focus	Function
1	Communication	<ul style="list-style-type: none"> ● Model utilization and execution ● Data and information sharing ● Model access and retrieval control
2	Modeling and Negotiation	<ul style="list-style-type: none"> ● Model sharing ● Data-model integration ● Model integration ● Model modification and creation ● Model selection and evaluation
3	Intelligent Support	<ul style="list-style-type: none"> ● Automated model integration ● Advising on model selection and evaluation ● System learning

models. Here, the creation of new models means construction of new models from scratch. Again, the model modification and construction processes may involve only a member or a group of members working together. In general, it requires a group model development language to support various modeling stages, from model specification to verification and validation. Similar to that in an individual MMS, the language may be composed of a model definition language for specifying requirements and a model manipulation language for structuring model components (Blanning, 1982; Dolk, 1986; Liang, 1985).

Finally, the system supports model evaluation. With the help of GMMS, group members can examine model assumptions, compare the accuracy of different models in different situations, discuss criteria and procedures for model evaluation, actually evaluate models, and disseminate the evaluation results.

The third-level GMMS focuses on intelligent support. Major functional capabilities include automated model integration, advising on model creation and selection, and system learning. The major difference between this level and the second level is the degree of automation. For example, the second-level model integration capability requires that group members find proper component models in the model base and schedule the sequence of execution by a manual process involving discussions and trial-and-error; whereas model integration at the third level will try to locate component models based on the specifications provided by the users and will suggest proper sequence of execution automatically. Advising on model selection helps group members identify criteria for comparison and select an acceptable model. System learning capabilities tailor the GMMS to the requirements of individual group members. For example, when members in the group have different preferences in information presentation format, the system can present the same model outputs to different members in different formats (Liang, 1987). Furthermore, if necessary, different models can be recommended to different members according to their previous usage records.

Decision environment

The third dimension portraying GMMS design is the environment in which group decisions are made. An individual MMS typically supports two types of model utilization: single-user-single-

model and single-user-multiple-model. In addition to these, a GMMS needs to support multiple-user-single-model (MUSM) and multiple-user-multiple-model (MUMM) cases. In an MUSM situation, group members use a shared model to make decisions. In an MUMM situation, multiple models are used and shared by group members. Because either MUSM or MUMM involves multiple users who may be geographically centralized or dispersed, and the models to be used by the group may be owned by an individual member or the group, there are at least two possible environments in which model management support is provided: centralized and distributed.

A centralized GDSS includes three basic components: a user interface, a GMMS, and a group database management system (GDBMS). All users interact with the same system. Data are stored in the group database (GDB) and models are stored in the group model base (GMB). The GMMS controls all access to the individually owned and group-owned models in the model base. Figure 2 illustrates a centralized system design.

In a distributed environment, each user has access to an individual DSS and a GDSS. Each individual DSS contains models and data owned by a particular user. The GDSS maintains a GMB and a GDB. The GMB stores models developed and owned by groups and models designed for supporting group activities such as nominal group techniques and brainstorming processes. It also coordinates MMSs of various individual DSSs and maintains indexes to all models stored in individual model bases. Figure 3 illustrates a distributed GDSS.

During system use, the difference between these two types of systems should be transparent to the users. In other words, a distributed system should allow members to access models stored in an individual model base without need to go through a tedious, manual search process. For example, when user *i* places a request to examine the model user *j* used to forecast sales for next year, the GMMS should pull out the model from user *j*'s model base (upon the approval of user *j*), store it in the GMB temporarily, and present the model to user *i*. In this case, user *i* interacts with the GDSS only and does not need to take care of the distributed storage.

In summary, although quantitative models are usually designed for solving well-structured problems, GMMS can benefit both the structured and

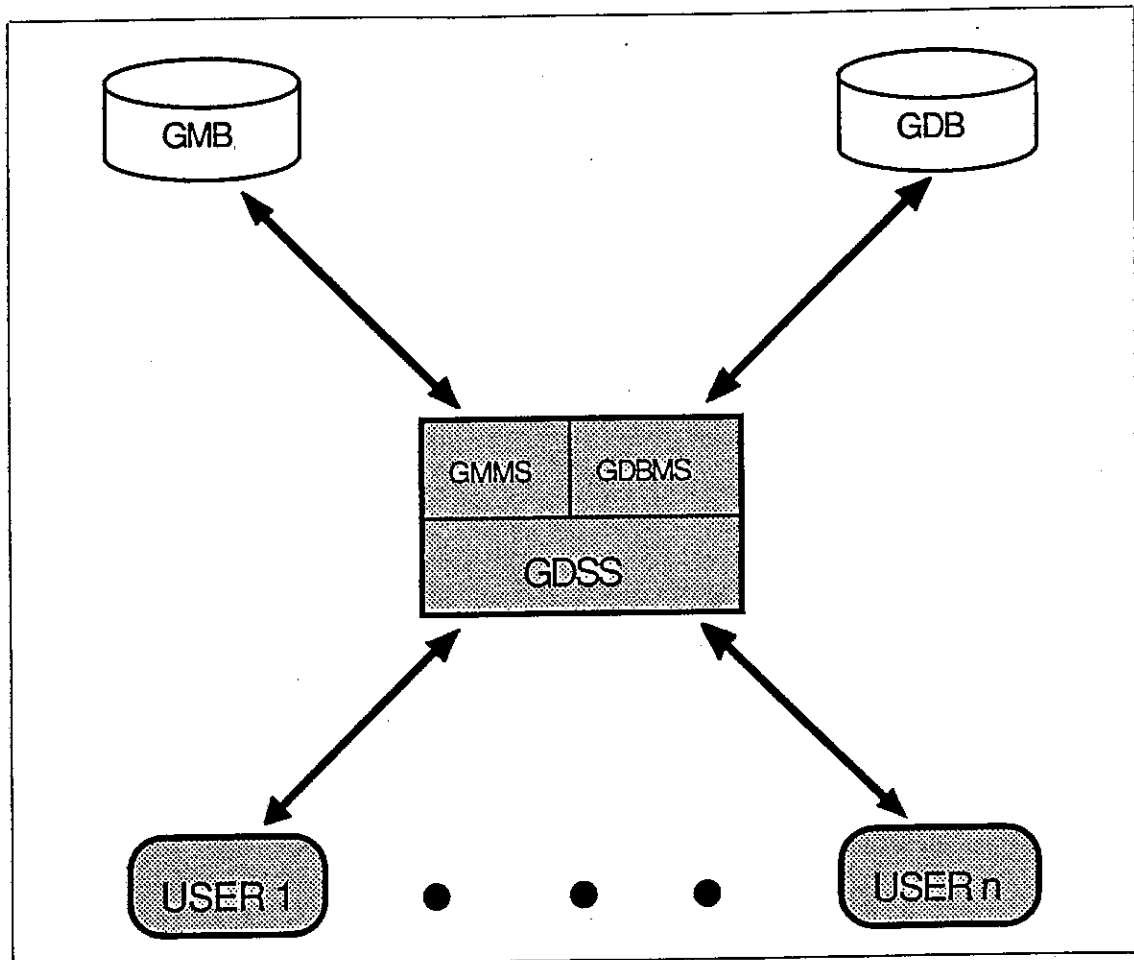


Figure 2. A Centralized System Design

unstructured tasks in group decision processes. Since different tasks and groups need different levels of support, GMMS functions can be divided into three levels, and the decision support environment can be either centralized or distributed.

Design of Group Model Management Systems

Given the understanding of group model management requirements, the next question would focus on how this kind of system can be designed and implemented. In this section, a knowledge-based architecture capable of delivering the desired GMMS support is presented and several issues involved in implementing the framework are discussed. The architecture expands existing GDSS frameworks and individual MMS literature. Since level-three systems are gener-

ally a superset of level-two and level-one systems, discussions will center around this level.

GMMS architecture

The model management functions described in the previous section fall into three general categories: model utilization, model development, and system adaptation. In order to provide all these functions, therefore, a GMMS needs to have at least four subsystems: a model utilization subsystem, a modeling subsystem, a learning subsystem, and an inference engine. Figure 4 illustrates a conceptual architecture for such a system. The model utilization subsystem concentrates on effective use of decision models; whereas, the modeling subsystem focuses on facilitating model integration and construction. The learning subsystem collects information from users and outside environments in order to evolve the system. The inference engine inte-

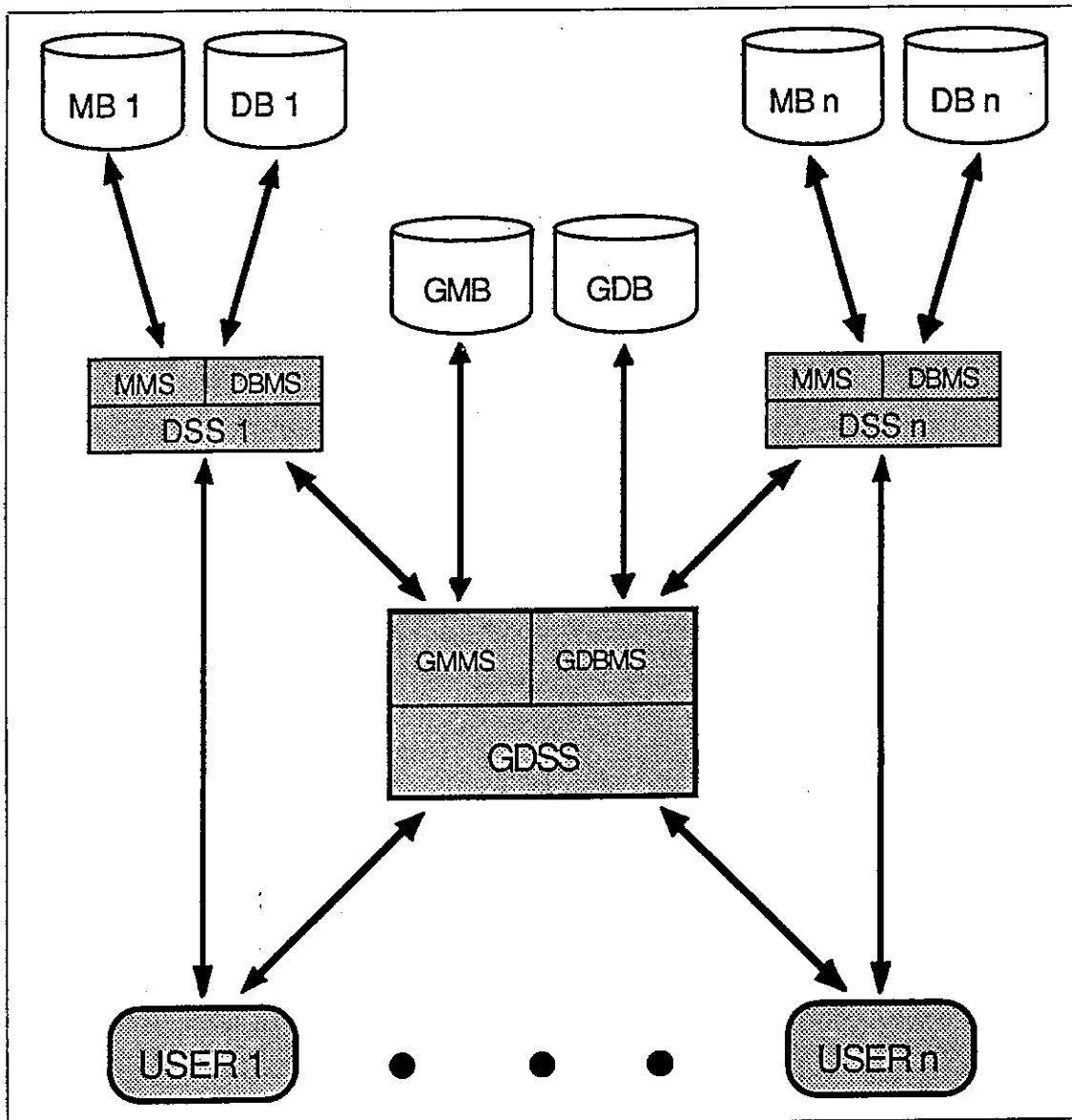


Figure 3. A Distributed System Design

grates all modules in the system and drives the processes of using and developing models. Knowledge base, model base, and database are repositories of relevant knowledge, decision models, and data, respectively. The correspondence between subsystems and the desired model management functions are briefly illustrated in Table 2.

Given current technologies in DSS, local area networks, and distributed data processing, development of a GMMS with some of the level-one and level-two functions is technically feasi-

ble. In fact, prototype implementations do exist. For example, the PLEXSYS system developed at the University of Arizona supports many model management functions including selection and initialization of models by planning session facilitators and execution of models by planners (see Applegate, et al., 1987 and Konsynski, et al., 1984 for a detailed description of the system). In addition, some prototypes developed for individual MMS, such as TIMMS (Liang, 1988b), can also be expanded to support group meetings. Since a complete discussion of all technical details in GMMS design is beyond the

scope of this article, the following sections will focus on the skeleton and will refer to available, relevant literature that provides more information.

Model Utilization Subsystem

The model utilization system may be the most frequently used module. Group members inter-

act with this subsystem to retrieve and execute selected models. Four functions are essential to effective support of model utilization: meeting control, query processing, report generation, and help.

The meeting control function schedules multiple requests for access to a particular model and

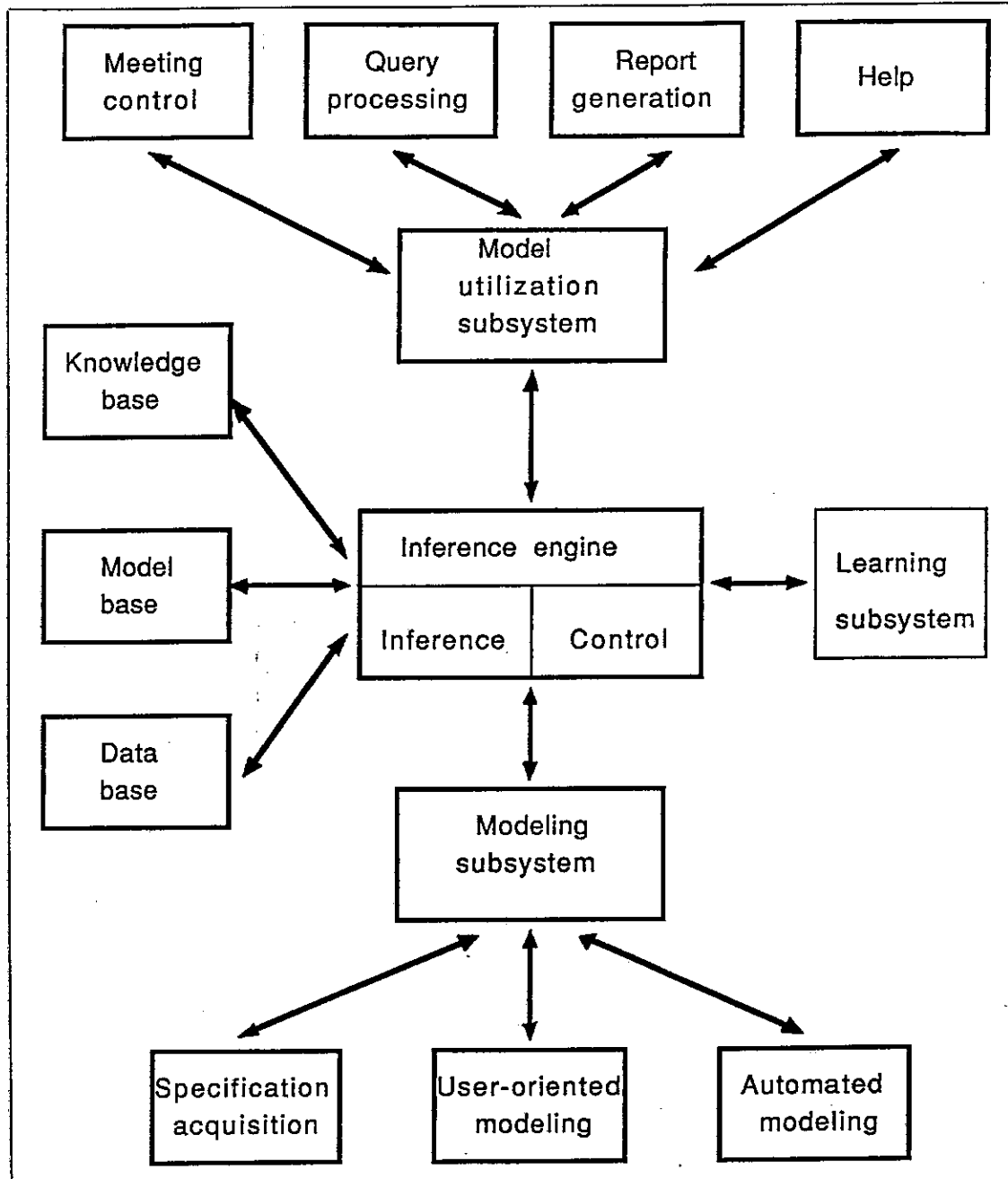


Figure 4. A GMMS Architecture

Table 2. Model Management Functions Supported by System Modules

System Module	Model Management Function
Model Utilization	<ul style="list-style-type: none"> ● Model execution ● Data and information sharing ● Model access and retrieval control ● Model sharing ● Model selection and evaluation
Modeling	<ul style="list-style-type: none"> ● Model construction ● Data-model integration ● Model modification ● Model integration
Learning	<ul style="list-style-type: none"> ● System learning and adaptation ● Automated modeling
Inference Engine	<ul style="list-style-type: none"> ● Subsystem integration and agenda control

prepares an agenda for communication among group members. In addition, when access to an individual MMS is needed in a distributed GMMS, it contacts the MMS, obtains copy permission, and then pulls out the model and saves it in the group model base for use. In order to handle situations where several requests are placed simultaneously, mechanisms must be designed to avoid deadlock and to control system concurrency. The session controller in PLEXSYS, for example, is a sample implementation of the meeting control module.

The query processing function serves as an interface through which the users place their requests. The module translates a user query into machine-understandable commands of the system. A natural language interface or implementation of a structured query language, such as SQL in database management, is acceptable (see Blanning, 1984 and Liang, 1988b for sample implementation).

The report generation function presents output information produced by a model to group members. Since different members may prefer different formats, this module must be able to tailor its presentation to individual requirements. This can be achieved by either providing a report generation language in which the user specifies the desired format or learning from the users' previous choices. In the latter case, learning capabilities, which will be discussed later, are essential.

The help function provides group members with necessary information for using the GMMS and information regarding a particular model in use.

One critical role of the help module in group decision making is presenting model assumptions, limitations, and internal structure to users when there is need for detailed examination of models. This frequently happens in the processes of model comparison, evaluation, selection and integration. In general, graph-based model representation and manipulation mechanisms (e.g., Geoffrion, 1987; Liang, 1988a; 1988b) are appropriate for rendering model structures, and frame-based mechanisms (e.g., Dolk and Konsynski, 1984) are capable of presenting model assumptions, constraints, and other implementation details.

Modeling Subsystem

The modeling subsystem provides an environment for developing or modifying a model in a group meeting. It must support at least three functions: specification acquisition, user-assisted modeling, and automated modeling.

The specification acquisition function helps group members obtain accurate information regarding the problem faced by the group. The desired information may include model assumptions, integrity constraints, decision variables, causal relationships among variables, and available alternatives. This information can be used to choose a proper model from the model base or to develop a new model. Of course, the information must go through a full group discussion before it can be used to develop models.

After obtaining specifications, the group can either develop a model manually or ask the system to formulate a model automatically. If the

former is chosen, the group will use a modeling language provided by GMMS to construct the new model. This is called user-assisted modeling. The most critical issue in supporting user-assisted modeling is to design a modeling language with an adequate set of primitive functions. Previous research in modeling and modeling languages, such as Geoffrion's (1987) structured modeling framework, may provide a starting point for this line of work. A very good review of 41 tools for structural modeling can be found in Lendaris (1980).

If automated modeling is desired, the system will formulate an ad hoc model based on the defined model specification and available tools. Automated modeling is much more difficult than user-assisted modeling. Given current information technology, design of a complete automated modeling system that can create models from scratch is almost infeasible. One major difficulty is that a modeling process usually involves a huge amount of common sense — a set of knowledge computers cannot yet handle. A feasible goal at present is to develop new models by integrating existing models. In other words, the focus would be model integration. Although this is imperfect, it can still provide valuable support. For example, when a group needs a model to examine the relationship between forecasted sales and production schedules, the GMMS can link existing sales forecasting and production scheduling models. This saves the effort of creating a brand-new model and still provides a satisfactory solution to the problem.

Several different approaches have been proposed for handling model integration (Blanning, 1982; 1986; Elam, et al., 1980; Geoffrion, 1987; Konsynski and Dolk, 1982; Liang, 1988a, 1988b). One approach is to consider each existing model as a primitive operator and consider the automated modeling process as a planning process by which a set of operators can be found and scheduled to eliminate the difference between the goal state and the initial state. In this approach, the goal state contains the resulting model that can produce the desired information; whereas, the initial state includes all operators and input data. By these definitions, a compound model formulated by the automated modeling mechanism is a macro-operator — a sequence of primitive operators. The major research issue is to develop effective mechanisms that can integrate operators into appropriate macro-operators (Korf, 1985; Liang, 1988a).

Learning Subsystem

System learning is an advanced capability for GMMS. It allows the system to take advantage of its own experience and hence to improve its performance when a similar problem is encountered later. Basically, a GMMS learns in two ways. First, the system learns model development and model selection by examining previous uses of models. This helps the modeling subsystem to automate the modeling process and to provide advice regarding model selection. Second, the system learns users' preferences by examining their previous choices. This may improve user friendliness and facilitate negotiation. Both require the learning subsystem to keep track of previous usage patterns and adapt the system accordingly.

Although machine learning is still an experimental area in artificial intelligence, GMMS can handle modeling by using the macro-operator approach to learn previously developed or selected models (Korf, 1985). Once a macro-operator has been formulated for a particular group task, the GMMS can store the macro-operator in the model base. When a similar task is encountered later, the system directly retrieves it without need to go through another modeling process.

The second task handled by the learning subsystem is to learn individual user's preferences (Liang and Jones, 1987). This can be done by keeping track of a user's previous usage. For example, suppose different members in a group prefer different sales forecasting models — one prefers linear regression and another prefers exponential smoothing. With this information the coordinator of the meeting can anticipate some sort of conflict in the meeting and the GMMS is then able to help both the coordinator and group members prepare in advance. When a conflict on sales forecasting actually occurs during the meeting, GMMS can make this information available to group members to facilitate negotiation or even suggest alternatives for resolution if there is enough evidence. In addition to simply recording previous usage, GMMS can also infer future uses from previous data. For example, self-adaptive mechanisms can be implemented to increase user friendliness by changing the presentation formats when user preference evolves (Liang, 1987).

Inference Engine

The inference engine is the heart of a GMMS. It integrates all subsystems and performs two

important functions: inference and control. Four mechanisms are essential for inference and control: (1) integrating the model base and the database, (2) integrating models in the model base for automated modeling, (3) controlling the execution of selected models, and (4) controlling system learning and adaptation.

The first mechanism retrieves data required for model execution from the database. The second mechanism allows models in the model base to be integrated to form a complex model. It serves as a basis for automated modeling. The third mechanism schedules group activities when multiple requests on model execution or modeling are presented. The last mechanism determines what knowledge should be acquired by the system and how the system can tailor itself to satisfy group members with different requirements.

The inference engine also links the other three modules: *knowledge base*, *model base*, and *database*. The knowledge base maintains all knowledge relevant to GMMS, including model creation, meeting control, system learning, user preference, and system usage. The model base contains decision models and models that support group techniques such as brainstorming and nominal group methods. The database stores data pertaining to decision making and data pertaining to system use and control.

Implementation issues

The architecture presented previously provides a framework for GMMS design. Implementation of the framework, however, is not without problems. There are several technical and behavioral issues to be overcome. Although existing technologies are adequate for prototyping some primitive GMMS functions, technical problems usually appear when attempts are made to add advanced capabilities such as automated modeling and system learning. In addition, the more powerful a GDSS is, the deeper a group decision process would be interrupted. This can take group members longer to learn the system and generate stronger resistance. Some of the issues are discussed in this section. Since most of these issues are complex and need further research, this discussion should be considered an initiation rather than a conclusion.

The first issue that may arise in using GMMS is why model sharing should be encouraged. The basic philosophy of model sharing is that it can improve the communication and negotia-

tion processes by reducing uncertainty resulting from incomplete information. Model sharing, however, is not without negative effects — to some members they lose control over their valuable resources, which eventually would result in reduced influence (or power) in the group. Therefore, clear guidelines for model sharing must be established to avoid potential problems.

The second issue pertaining to implementation is how to assess whether a particular system is adequate for a task. Although GMMSs are divided into three distinct levels in the article, they are actually a continuum in the real world. Determining a proper combination of functions must at least consider the nature of the tasks to be supported, the expectation and background of the group members, the technical capabilities of the development group or vendors, and the financial resource available for the particular project. It is not necessarily true that the more functions a GMMS provides, the better the group performance will be. Sometimes, extra functions can be a heavy burden for the group if not properly implemented.

If a GMMS needs to incorporate advanced capabilities, then one problem facing system designers is the availability of techniques for implementing these capabilities efficiently. Although we have briefly described several applicable approaches to automated modeling and system learning, most of them are experimental in nature, and it may take decades before those technologies become mature. Even if satisfactory mechanisms for automated modeling are developed, another issue would raise the question of whether the users trust the created model, i.e., how the model generated by the system can be evaluated appropriately. What kinds of criteria are appropriate for evaluation? Who is going to evaluate the model? If different group members have different evaluations on the developed model, then whose evaluation should be accepted or how can this conflict be resolved? Can the GMMS support this conflict resolution? All these problems indicate that there is plenty of potential research in this important area.

Concluding Remarks

Given the increased interest in applying information technology to support group problem solving, this article has examined various issues in designing and implementing GMMS. First, the difference between group and individual deci-

sion processes requires GMMS to provide functions that support group activities such as negotiation and conflict resolution. These functions range from model and data sharing to intelligent model construction, and can be classified into three general categories: communication, modeling and negotiation, and intelligent support. Their implementation must consider the scope of the system and the decision environment.

In order to deliver the required capabilities, a software architecture is then presented. It includes an inference engine and three major subsystems. The model utilization and modeling subsystems support communication, modeling, and negotiation requirements. The learning subsystem provides intelligent capabilities for automation and system adaptation.

Like many other large-scale projects, this work is by no means final. Three implementation problems that need more research have been discussed. First, modeling sharing may change the power structure in a group and hence result in resistance. Second, it is not necessary that the more functions a GMMS supports, the better the system is. The designer needs to make judicious tradeoff among various factors. Finally, technology for implementing some of the advanced group model management functions is not yet available, which indicates promising areas for future research.

Acknowledgements

The author would like to thank Benn Konsynski and anonymous reviewers for their helpful comments on earlier versions of the article. The work was partially supported by a Campus Research Board Award and a summer research grant from the Department of Accountancy.

References

- Applegate, L.M., Konsynski, B.R. and Nunamaker, J.F. "Model Management Systems: Design for Decision Support," *Decision Support Systems* (2:1), March 1986, pp. 81-91.
- Applegate, L.M., Chen, T.T., Konsynski, B.R. and Nunamaker, J.F. "Knowledge Management in Organizational Planning," *Journal of MIS* (3:4), Spring 1987, pp. 20-38.
- Blanning, R.W. "A Relational Framework for Model Management in Decision Support Systems," *DSS-82 Transactions*, 1982, pp. 16-28.
- Blanning, R.W. "Conversing With Management Information Systems in Natural Language," *Communications of the ACM* (27:3), March 1984, pp. 201-207.
- Blanning, R.W. "An Entity-Relationship Approach to Model Management," *Decision Support Systems* (2:1), March 1986, pp. 167-172.
- Bonczek, R.H., Holsapple, C.W. and Whinston, A.B. "The Evolving Roles of Models in Decision Support Systems," *Decision Sciences* (11), April 1980, pp. 337-356.
- Bui, T. and Jarke, M. "Communication Requirements for Group Decision Support Systems," *Journal of MIS* (2:4), Spring 1986, pp. 8-20.
- Chang, E. "Participant Systems," *Future Computing Systems* (1:3), 1986.
- Decker, K.S. "Distributed Problem-Solving Techniques: A Survey," *IEEE Transactions on Systems, Man, and Cybernetics* (SMC-17:5), September/October 1987, pp. 729-740.
- DeSanctis, G. and Gallupe, R.B. "Group Decision Support Systems: A New Frontier," *Data Base* (16:2), Winter 1985, pp. 3-10.
- DeSanctis, G. and Gallupe, R.B. "A Foundation for the Study of Group Decision Support Systems," *Management Science* (33:5), May 1987, pp. 589-609.
- Dolk, D.R. "Data as Models: An Approach to Implementing Model Management," *Decision Support Systems* (2:1), March 1986, pp. 73-80.
- Dolk, D.R. and Konsynski, B.R. "Knowledge Representation for Model Management Systems," *IEEE Transactions on Software Engineering* (SE-10:6), November 1984, pp. 619-628.
- Dutton, W.H. and Kraemer, K.L. "The Politics of Modeling: Computer Models in the Policy Process," *Systems, Objectives, Solutions* (3:1), 1983, pp. 13-29.
- Elam, J.J. and Konsynski, B.R. "Using Artificial Intelligence Techniques to Enhance the Capabilities of Model Management Systems," *Decision Sciences* (18:3), Summer 1987, pp. 487-502.
- Elam, J.J., Henderson, J.C. and Miller, L.W. "Model Management Systems: An Approach to Decision Support in Complex Organizations," *Proceedings of the First International Conference on Information Systems*, Philadelphia, PA, December 8-10, 1980, pp. 98-110.
- Fisher, B.A. *Small Group Decision Making: Communication and Group Process*, McGraw-Hill, New York, NY, 1974.
- Gallupe, R.B., DeSanctis, G. and Dickson, G.W. "Computer-Based Support for Group Problem-Finding: An Experimental Investigation," *MIS Quarterly* (12:2), June 1982, pp. 276-296.

- Geoffrion, A.M. "Introduction to Structured Modeling," *Management Science* (33:5), May 1987, pp. 547-588.
- Goncalves, A. "Group Decision Methodology and Group Decision Support Systems," *DSS-85 Transactions*, San Francisco, CA, April 1-4, 1985, pp. 135-142.
- Gray, P. "Group Decision Support Systems," *Decision Support Systems* (3:3), September 1987, pp. 233-242.
- Huber, G.P. "Issues in the Design of Group Decision Support Systems," *MIS Quarterly* (8:3), September 1984, pp. 195-204.
- Jarke, M. "Knowledge Sharing and Negotiation Support in Multiperson Decision Support Systems," *Decision Support Systems* (2:1), March 1986, pp. 93-102.
- Klein, G. "Developing Model String for Model Management," *Journal of MIS* (3:2), Fall 1986, pp. 94-110.
- Konsynski, B.R. and Dolk, D.R. "Knowledge Abstractions in Model Management," *DSS-82 Transactions*, 1982, pp. 187-202.
- Konsynski, B.R., Kottemann, J.E., Nunamaker, J.F. and Stott, J.W. "PLEXSYS-84: An Integrated Development Environment for Information Systems," *Journal of MIS* (1:3), Winter 1984, pp. 64-104.
- Korf, R.E. "Macro-Operators: A Weak Method for Learning," *Artificial Intelligence* (33), 1985, pp. 35-77.
- Lendaris, G.G. "Structural Modeling — A Tutorial Guide," *IEEE Transactions on Systems, Man, and Cybernetics* (SMC-10:12), December 1980, pp. 807-840.
- Liang, T.P. "Integrating Model Management with Data Management in Decision Support Systems," *Decision Support Systems* (1:3), September 1985; pp. 221-232.
- Liang, T.P. "A Graph-Based Approach to Model Management," *Proceedings of the Seventh International Conference on Information Systems*, San Diego, CA, December 1986, pp. 136-151.
- Liang, T.P. "User Interface Design for Decision Support Systems: A Self-Adaptive Approach," *Information & Management* (12:4), April 1987, pp. 181-193.
- Liang, T.P. "Reasoning in Model Management Systems," *Proceedings of the Twenty First Hawaii International Conference on Systems Sciences* (Vol. III), Kona, HI, January 1988a, pp. 461-470.
- Liang, T.P. "Development of a Knowledge-Based Model Management Systems," *Operations Research*, 1988b, forthcoming.
- Liang, T.P. and Jones, C.V. "Design of A Self-Evolving Decision Support Systems," *Journal of MIS* (4:1), Summer 1987, pp. 59-82.
- Liang, T.P. and Jones, C.V. "Meta-Design Considerations in Developing Model Management Systems," *Decision Sciences* (19:1), Winter 1988, pp. 72-92.
- Nunamaker, J.F. Jr., Applegate, L.M. and Konsynski, B.R. "Facilitating Group Creativity: Experience with a Group Decision Support System," *Journal of MIS* (3:4), Spring 1987, pp. 5-19.
- Rathwell, M.A. and Burns, A. "Information Systems Support for Group Planning and Decision Making Activities," *MIS Quarterly* (9:3), September 1985, pp. 255-271.
- Simon, H.A. *The Science of the Artificial*, MIT Press, Cambridge, MA, 1981.
- Sprague, R.H. Jr. and Carlson, E.D. *Building Effective Decision Support Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
- Sprague, R.H. Jr. and Watson, H.J. "Model Management in MIS," *Proceedings of the 7th AIDS Meeting*, 1975, pp. 213-215.
- Stohr, E.A. and Tanniru, M.R. "A Database for Operation Research Models," *International Journal of Policy Analysis and Information Systems* (4:1), June 1980, pp. 105-121.
- Vogel, D., Nunamaker, J.F., Applegate, L.M. and Konsynski, B.R. "Group Decision Support Systems: Determinants of Success," *DSS-87 Transactions*, June 1987, pp. 188-128.
- Will, D.J. "Model Management Systems," in *Information Systems and Organization Structure*, E. Grochla and N. Szyperski (eds.), Walter De Gruyter, Berlin, 1975, pp. 467-482.

About the Author

Ting-Peng Liang is an assistant professor in the College of Commerce and Business Administration at the University of Illinois, Urbana-Champaign. He received an MBA from National Sun Yat-sen University (in Taiwan, Republic of China), and MA and Ph.D. degrees in information systems from The Wharton School of the University of Pennsylvania. His research interests include model management, decision support systems, applied artificial intelligence, and strategic information systems.