Faculty and Researchers | Faculty and Researchers' Publications

1985

Model management in organizations

Dolk, Daniel R.; Konsynski, Benn R.

http://hdl.handle.net/10945/48251

SOS

Model Management in Organizations

Daniel R. Dolk
Naval Postgraduate School, Code 54DK, Monterey, CA 93943, USA

and

Benn R. Konsynski
The University of Arizona, Tucson, AZ 85721, USA

The premise that the personal computer/spreadsheet explosion will result in the evolution of model management within organizations is explored. The authors use Nolan's stage model of organizational data processing activity as a basis for discussing the nature of change in organizations as local computing capability proliferates. The mainframe era resulted in the recognition of data as a resource and gave rise to data administration. The authors expect that the personal computer era and the accompanying spreadsheet explosion will lead to the recognition of models as a valuable and manageable resource. The role of model administration within organizations is discussed as are software tools for supporting this functional activity. The information resource encyclopedia, an extension of the traditional data dictionary concept, and the model management system are introduced as integral components for supporting model administration. An example is presented to suggest an integrative approach for implementing an MMS in a spreadsheet environment.

Daniel R. Dolk received the Ph.D. degree in management information systems from the University of Arizona, Tucson, in 1982. Since 1982, he has been with the Department of Administrative Sciences, Naval Postgraduate School, Monterey, CA, where he is currently an Assistant Professor. His research interests include model management systems, decision support systems, and the application of artificial intelligence to information administration. Dr. Dolk is a member of the Association for Computing Machinery, the American Association for Artificial Intelligence, the Institute of Management Sciences, and the IEEE Computer Society.

1. Introduction

The advent and widespread acceptance of microcomputer spreadsheet programs has given rise to some interesting problems within organizations. Spreadsheets have disseminated modeling far and wide to every nook and cranny of the organization. They have been far more effective in introducing modeling concepts and motivating people to build and use models than any previous device or management approach. It is not an exaggeration to claim that spreadsheets have put models and powerful modeling capabilities into the hands of unskilled modelers.

The utility of spreadsheet programs has been well-documented and is certainly reflected in the marketplace. For the time being, let us characterize these advantages under the rubric of "increased productivity". On the downside, however, the rampant proliferation of spreadsheets has caused major headaches for management, especially data processing management. The sudden introduction and unchecked growth of this new information resource has left many organizations uncertain (some even paralyzed) about how to manage and control this phenomenon. As a result, perceptions of management are beginning to change. The advantage of increased individual productivity resulting from the use of modeling

Benn R. Konsynski received the Ph.D. degree in computer science from Purdue University, Lafayette, IN, in 1976. He is currently an Associate Professor in the Department of Management Information Systems, University of Arizona, Tucson. He has worked extensively on the development of computer aids in systems requirements specification and analysis and in automation of software design and code generation. He is currently working on development of computer aids in distributed systems and office systems design, interorganizational systems, and model and dialog management.

must be weighed against the organizational disequilibrium caused by lack of control of this function.

The purpose of this paper is to examine the potential of a new organizational function, model management, and its organizational opportunities. Model management recognizes models as a corporate resource which much be controlled and managed like any other resource. The need is not unlike that associated with the need for data management. Model management can be viewed as a direct corollary to data management. In particular, the role of a model administrator (MA) is introduced and its functions likened to those of a data administrator (DA). A model management system (MMS) and information resource encyclopedia (IRE) are discussed as software systems which serve the objectives of model administration just as a database management system (DBMS) does for the DA. Finally, an integrative strategy for implementing an MMS and IRE in an existing spreadsheet environment is presented to demonstrate how these systems support model organization.

2. Brief History of Model Management

Before the appearance of spreadsheets, it was very difficult to get users to recognize the formal models that were frequently in use in organizational decision-making. One of the reasons for this was the high cost of model development. Figure 1 shows the results of a 1974 National Science Foundation survey of methematical models developed within the U.S. government [6]. This survey revealed that the average cost of developing one equation was $6000 and three weeks of effort while the eventual use of the models rarely migrated to the policymaking level. Formal modeling was perceived as a very costly activity. One reason for this condition was, and in many cases continues to be, the reluctance of many managers to use models they have difficulty understandig and interpreting. This was compounded by the obtuseness of second and third generation modeling software which tended to obfuscate, rather than clarify, what was going on in a model. This intersection of models and computers raised a technological barrier which severely hampered the use of models in many organizations. Model

Sample Size	650 models
Total Cost	$100,000,000
Average Development	17 months
Median Model Size	25 equations

Average cost to develop 1 equation:
 3 weeks and $6000

75% of models can be operated only
 by the original development team

Use of models in policymaking minimal

Fig. 1. NSF Model Development Survey (as reported in [1]).

management and model management systems [11] were suggested as a means of overcoming these barriers by explicitly recognizing modeling as an organizational function subject to planning, control and operation.

The advent of spreadsheets has changed this attitude towards modeling in a very short period. Suddenly people at all levels of the organization are building models and using them in making decisions. Not only has the longstanding modeling inertia described above been overcome, it has even become de rigeur to do modeling. This surprising turnaround bears witness to the fact that one of the major obstacles to modeling has been the clumsy software vehicles previously available for performing this function. The proliferation of models resulting from spreadsheets has led, however, to serious problems in managing and controlling this new information resource. These problems will undoubtedly require some form of managerial control, i.e. model management, for their resolution.

3. Modeling and the Evolution of Data Processing

One useful way of viewing this phenomenon is in the context of Nolan's model of the evolution of the data processing organization. An early version of this model [9] hypothesizes four stages of evolution: initiation, contagion, control, and integration. As a computer resource is introduced into an organization (initiation) and gains acceptance and use (contagion), management realizes a need for control (control) and eventually managing and planning for this resource (integration). Thus, re-

sponsibility for managing the resource migrates higher and higher in the organization as the evolution proceeds (Figure 2). The initiation and contagion stages focus on the operational level, the control stage marks a critical transition point where management consciously recognizes the need for controlling the contagion phenomenon, and the integration stage assimilates the resource management into the overall organizational process.

This model was originally proposed to explain the evolution of mainframe data processing within organizations. For those who were present in the hectic 60's, the appearance of computers in various departments and their inevitable upgrade and migration to a centralized installation is a familiar pattern which seems to fit the evolution model well. The curious facet of this model is the illusory integration stage which never seems to be attained because of the tendency of this model to cycle with each new information resource. Nowhere is this more apparent than the personal computer revolution which is faithfully going through the same evolutionary stages (currently probably in the contagion or early control stage) as mainframes but which nevertheless continues to baffle management and defy organizational control. This continual need to reintegrate each new resource seems unnecessarily inefficient and suggests another stage of evolution which is more strongly planning-oriented.

In a later version of this model [10], Nolan introduces two such stages: data administration and maturity. Data administration refers to an organization's perception that data is an organizational resource and must be managed as such. This marks a shift away from managing hardware to managing information resources. Data administration corresponds to the data management approach which earmarked the 70's. Maturity is to the 6-stage model what integration is to the 4-stage version, namely an equilibrium resulting from the total assimilation of the information processing activity into the organization at all levels. This is closely related to what we call information resource management today.

In examining the spreadsheet phenomenon, it is safe to say that spreadsheets are probably in the late contagion or early control phases of evolution. There has been a rapid proliferation of spreadsheet and modeling packages in companies so one can argue that the contagion stage has been reached. If this trend mirrors the earlier evolution of mainframes, then we can expect to see the contagion stage give way to a growing awareness of the need for controlling the model resource. Initially this may be manifest as an attempt to control the microcomputer resource itself by setting policies for micro acquisition and use. The integration stage would involve the coordination of micros with existing hardware via distributed computing technology such as local area networks. This is already taking place in many companies today. Paralleling this development, we can expect to see spreadsheets and modeling evolving in a fashion reminiscent of the evolution and acceptance of the data management approach to controlling the data
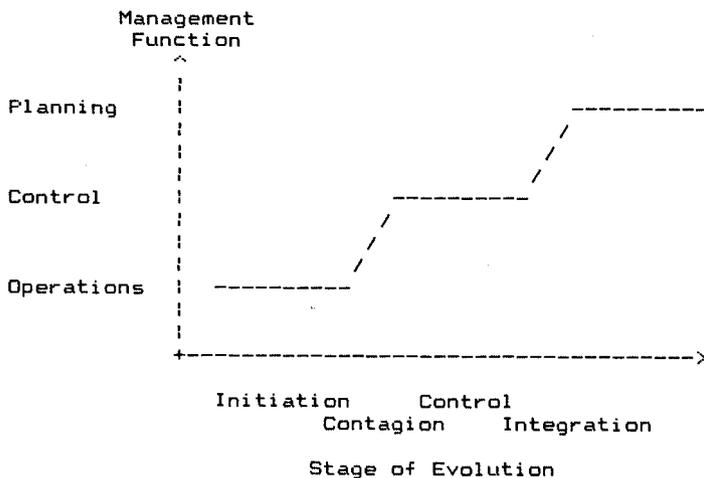

Fig. 2. Nolan's Model of Data Processing Evolution.

resource. We can expect that there will emerge a model administration stage of evolution that mirrors closely its data administration counterpart. As data management was born in the era of the mainframe environment, so may model management arise with the growth of the personal computing environment.

4. Need for Model Management

Data management arose in response to several inefficiencies regarding control and use of the data resource. The same problems that were evident in the recognition of the need for effective data management currently exist in the modeling and spreadsheet environment, e.g.:
redundancy and inconsistency,
integrity and security,
lack of sharing,
standardization,
physical/logical independence.

Redundancy in the modeling case refers to multiple versions of the same model developed independently of one another. Although, if planned for, different models may provide different and valuable insights about a particular problem, in general they tend to confuse rather than enlighten matters. Different assumptions underlying similar models can have a dramatic effect on how one interprets the results from those models. For example, in the military manpower environment, several models exist for calculating enlistment continuation rates, each based on overlapping but distinct assumptions. Proper use of this rate information requires an understanding of which model was used to calculate them. This oftentimes requires a significant amount of energy to detect and resolve inconsistencies. More frequently what happens is that this effort is foregone and model results are incorrectly applied.

Inconsistency among models raises questions of model accuracy, validity, and integrity. How does one evaluate and compare models in an environment where there is no guarantee that uniform data were used, for example? Furthermore, how does one uncover and analyze the underlying assumptions of a particular model? These are germane issues even in a disciplined modeling environment but they are particularly exacerbated when modeling activity is widely dispersed as is

the case with spreadsheets. This applies as well to the area of security when models use sensitive data or are themselves sensitive. It is understandably much more difficult to enforce appropriate security mechanisms when modeling resources are distributed throughout the organization. A fundamental question in many environments is "how to determine which models are more sensitive to environmental and policy changes?" It is reasonable to suspect that in many companies, there exists a wealth of organizational information, models, and data seldom used by anyone other than modelbuilders.

Lack of resource sharing is a major shortfall in modeling environments. Spreadsheets purport to increase individual productivity but hidden in this approach lie significant organizational opportunity costs. Spreadsheet models developed in support of individual productivity tend to be throwaway commodities, built and used one time and then discarded. This is clearly suboptimal from an organizational standpoint. If a model has proven useful in a decision situation, then we may assume that it may prove useful in similar or recurring situations. Given the high relative cost of building models (see Figure 1), it makes more sense to modify existing models, if available, than to start from scratch. Equally important, models may have transfer value so that a model developed in one context may be applicable to different situations as well. For example, an investment portfolio model might also be useful in designing an approach to acquiring microcomputers. The de facto "instant obsolescence" of models which exists in uncontrolled spreadsheet environments is not cost efficient and tends to mitigate the intended effects of increased productivity. The ability to share models helps to serve both individual and organizational needs.

Standardization is an issue which is strongly correlated with redundancy and inconsistency. In the data domain standardization of data elements insures that all users are accessing the same data. If two programmers want to use ZIP-CODE in their individual programs, they both get the same data structure; if two modelers want TELE-PHONE-DEMAND data, they get the same data values. This same need arises in the case of modeling. Models, like data structures, are a statement of department or organization policy. We expect consistency in the application of that policy. Rather

than have six different models of continuation rates, select one model and make that the standard. Then if modelers want to access continuation rate data, they all get the same data values. Instead of each organizational department devising its own budget model and format, select a standard format which all departments must use. The various budget models are still within each department's control but can easily be consolidated for organizational purposes. Another fertile area for standardization is model documentation. For each model in the organization's model inventory, annotate the exogenous and endogenous variables as well as the key underlying assumptions. This allows potential users to get a quick snapshot of what the model purports to do plus it contributes to a uniform way of viewing models within an organization. Other benefits of standardization can be enumerated but they all flow from the concept of resource sharing and treating models as a corporate asset.

Another important issue that parallels the concerns in data management is the issue of logical vs. physical independence. In the database environment, independence refers to the ability to define data entities and relationships logically without having to worry about how the data are stored physically. The best example of this is the relational data model [3]. The corresponding feature in the modeling situation is the ability to define models in a uniform fashion without having to worry about physical implementation. Currently this is not the case in a diverse spreadsheet environment where different packages require different representations (IFPS and Lotus 1-2-3 for

example). Although interface packages appear every day for reconciling different micro spreadsheet/database products, this quickly leads to a bewildering array of software linkages which tends to confuse, rather than simplify the situation. What is needed is an integrative mechanism providing uniform, high level model descriptions which can then be transformed to the appropriate representation for the target spreadsheet package. This capability can be provided by a model management system.

In addition to the problems which parallel those in data management, there is also an issue associated with the coordination of models with other information resources. In particular it is obvious that the relation between models and data is vital. Since models are users of data, they must be able to use existing data management capabilities for accessing the necessary data resources. This is currently a serious problem in spreadsheet environments where data are usually hand-entered into models. This results in data redundancy and integrity problems when the centralized data source changes but these changes do not get reflected in the spreadsheet. What is needed to remedy this situation is a strong link between the data management and model management environments. In the spreadsheet case, this translates into the need for flexible and powerful downloading and uploading capabilities between micros and the organizational database environment, whether on mainframes or local file or database servers. Models also need to be able to access and to link with other models. If they are truly a sharable resource, then it should be possible to build more complex,

```
+-----+      +--------------------+         -----------------
!     !      ! Model Description  !        / Uniform Model \
!Users+----->!                    !  +---->!               !
!     !      !     Language       !  !     \ Representation/
+-----+      +--------------------+  !       -------+--------
   ^                                 !             !
   !                                 !             V
----+----                            !     +-----------------+
/ Model  \                           !     !      Model      !
!        !                           !     !                 !
\Solution /                          !     ! Transformation  !
 ---------                           !     +--------+--------+
   ^                                 !             !
   !                                 !             V
   !         +--------------------+  !       -----------------
   !         ! Specific Modeling  !  !      / Specific Model \
   +---------+                    !  !<----!                 !
             !      Package       !        \ Representation /
             +--------------------+         -----------------
```
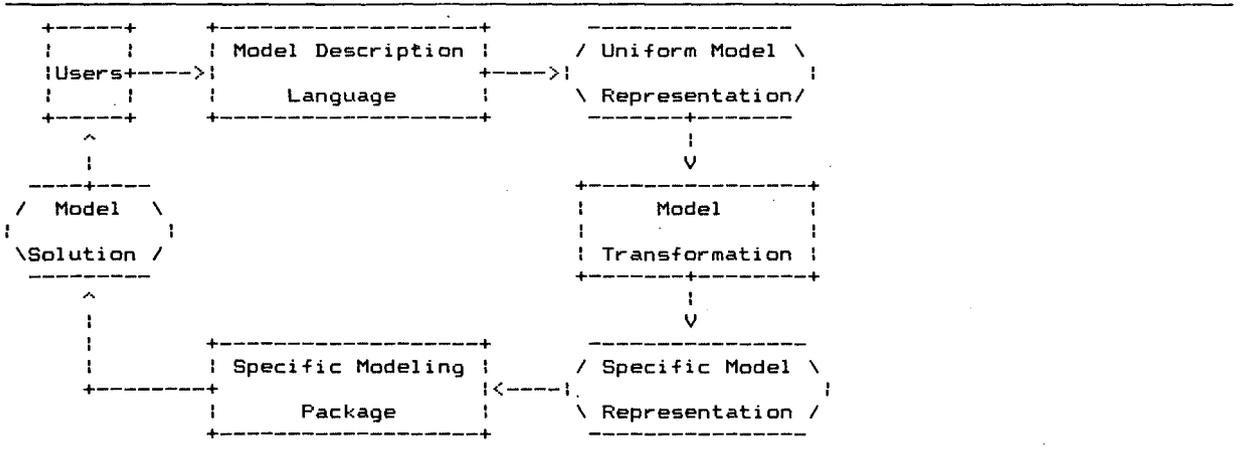
Fig. 3. Separation of Model Description and Model Solution.

composite models using results from one or more existing models. This presumes the ability to link mainframe models with one another, mainframes with spreadsheets, and spreadsheets with spreadsheets. Mainframe models have tended to be too large and complex to link together. Mainframe and spreadsheet modeling links are scarce since most spreadsheet packages do not have mainframe counterparts. Spreadsheet-to-spreadsheet linking is becoming prevalent as new packages appear regularly which allow one spreadsheet to "talk" to another. Unfortunately this tends to exacerbate the proliferation of modeling resources in a single user environment rather than mitigate it. It is difficult to relate models across spreadsheets not built with the same spreadsheet manager. This problem is only compounded when we introduce a variety of spreadsheet management tools.

The ability of models to communicate with one another and be linked together requires a uniform means of model representation as mentioned earlier. This implies that model description and model solution must be separate functions, or in other words, logical and physical representations must be independent (Figure 3). Thus, everyone describes models using the same description language but the representation is transformed and downloaded to an appropriate selected modeling facility for solution. This will be discussed in more detail in a later section.

5. Model Administration Function

It is clear that many of the problems which gave rise to the data management movement in the 70's are currently manifest in modeling environments today. This is especially true with regard to the spreadsheet explosion. The problems facing management concerning how to control this phenomenon are pressing ones. In light of Nolan's model, it is realistic to expect that solutions to these problems will take a form quite similar to that adopted in the data management case. Thus organizations will begin to recognize that models are a valuable resource requiring integrated operation, control, and planning. In order to institute this philosophy, new organizational functions will have to be defined and implemented. One such function is *model administration* which we propose

as a solution to many of the problems currently besetting modeling environments within organizations.

In very broad terms, the model administration mission may be stated as follows:

1. To promote the formal specification and use of models active in the planning, administration, management, and operation of the organization;
2. To provide overall control of, and responsibility for the design, creation, collection, formal specification, integrity, management, and administration of all organization models;
3. To coordinate data definitions with the Data Administrator in accordance with organization directives;
4. To provide tools for the collection, specification, organization, retrieval, maintenance, and synthesis of the model inventory;
5. To establish policy for the formulation and normative use of models on an organization-wide basis.

Hidden in this charter are some unspoken assumptions and some potentially difficult implementation problems. Foremost among the former is the implicit assumption that an organization already has some form of data administration in place. Trying to fashion model administration "from scratch" or without regard for existing data resources is a futile exercise. Existing data management structure will have a significant impact upon, and should provide useful guidelines for, establishing model management structure. In terms of implementation, trying to set and enforce standards for normative building and usage of models can be very sensitive politically. Nevertheless, management has found this expedient necessary, and sometimes sufficient, in controlling programming environments, for example. Similar measures in the modeling domain are also required.

The model administration mission given above is descriptive rather than prescriptive in trying to encompass all possible contingencies. If it is at all like data administration, however, the overall function will likely divide into two separate subfunctions: administrative and technical [8]. The administrative function, encapsulated in the Model Administrator role, is concerned with planning and control of the modeling activity. This involves managerial and policy affairs including the determination of end user (i.e., decision-maker) func-

Function	Responsibility
Model description, definition, and documentation	MA
Coordination with data resource	MA/MBA
Access, security, and integrity	MA/MBA
Operation, maintenance, and management	MBA
Monitoring and performance evaluation	MBA
Development of and compliance with standards	MA/MBA
Software procurement and vendor interface	MA/MBA
Liaison, consulting, and training	MA/MBA

Fig. 4. MA/MBA Functional Responsibilities.

tional requirements and subsequently what should be in an organization's model banks. Concurrent with this function is the technical effort relevant to the building, operation, and maintenance of a model management system (MMS). This activity is embodied in the Model Bank Administrator (MBA) role. The MA and MBA are direct analogs of the Data Administrator (DA) and Data Base Administrator (DBA) in the data management domain. Figure 4 delineates various functional responsibilities for model administration and indicates which of the two principal roles are involved.

6. Location of Model Administration within the Organization

It is presumptuous to prescribe where the model administration function should fall within existing organizational structures. Different situations will suggest different solutions depending upon the organizational circumstances and needs. Two questions frequently arise in regard to this issue, however:

1. Should the model administration function be in a data processing-oriented area?
2. Should the model administration and data administration functions be distinct?

The first question involves the reluctance of non-DP managers to relinquish control and resources to the DP arm of the organization. This battle has already been fought in the data management area.

The dependence on software for implementing data management capabilities, in particular the DBMS, makes it illogical and unwieldy to divorce the data administration effort from data processing. The same is true for model management. Solutions to the proliferation of micros and spreadsheet will undoubtedly be effected at the hardware and software levels. Distributed computing systems, especially local area networks, will likely emerge as one means of centralizing control while retaining the benefits of personal computing and individual productivity. Model administration will be intricately involved with this technology so it would be inefficient for it to be isolated from the ongoing data administration and data processing functions.

At the same time, it is clear that the ultimate measure of the utility of models to the organization lies in the management and application domain. It is dangerous to relinquish total control over model management to those "merely technically oriented". Since model management depends so strongly upon, and to a significant degree will be built from, data management, shouldn't the latter be expanded to include the former? This is undoubtedly a viable alternative to incorporating model administration, particularly in the initial stages of implementation. However, there are unique concerns relevant to the modeling activity which go beyond the purview of data administration. The most obvious is the need for modeling expertise, e.g. a firm understanding of manage-
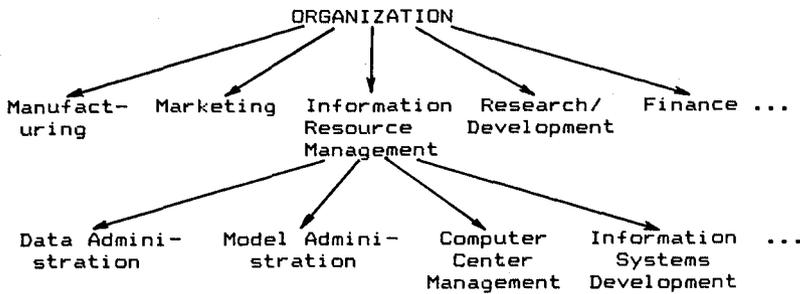
Fig. 5. Model Administration and Data Administration as functions of Information Resource Management.

ment science and operations research techniques and how they can be applied effectively to organizational decision-making. This knowledge is essential in developing model policy issues and strategy as well as shaping modeling standards and practice for an entire organization. Clearly, the skills required for this function transcend what is normally expected from the data administration role. This argues strenuously for a division of the two functions while maintaining close cooperation between them.

A more satisfying solution to the above problem is to incorporate model and data administration as sibling functions within an information resource management (IRM) environment (Figure 5). This identifies data and models as unique resources while recognizing their close relationship. More importantly it establishes a framework for the planning, operation, and control of *all* information resources rather than just hardware and software as implied in the earlier Nolan model. In fact, a fully operational IRM component earmarks an organization as "mature" in Nolan's terminology. Despite this idealized quality about IRM, it seems apparent that as more and more assets are identified as information resources, this evolution will become more and more of a reality. The recognition of models as an information resource is a vital step in this direction.

7. Software Tools for Model Administration

The implementation of model administration will rely heavily on software tools. Two key components are necessary for success: a dictionary/directory system (DDS) and a model management system (MMS). These correspond to the data dictionary/directory (D/D) and database manage-

ment system (DBMS) in the data administration environment. The DDS is primarily a tool for the MA which provides information about modeling resources whereas the MMS which stores, manipulates, and controls models is primarily the concern of the MBA. Both components may be part of an overall MMS depending on the level of integration it provides.

The dictionary/directory system may be thought of as a highly structured knowledge base of an organization's information resources. It contains information about data, models, hardware, programs, users, and other pertinent resources. The dictionary also contains information concerning the various interactions that occur among the data and processing entities. DDS's are most frequently discussed in the context of data, hence the familiar term "data dictionary", but this is an unnecessarily restrictive view. Data is only one information resource and by no means should be the only one contained within a DDS. The advent of model administration creates an equally vital need for including model information as well. Even the term "dictionary" in DDS is outmoded in that it conjures up an image of definitions arranged in alphabetical order. A more suitable term is "encyclopedia" with the intimation of a wide array of knowledge about a subject above and beyond simple definitions [7]. Thus we might speak more satisfactorily of an information resource encyclopedia (IRE) instead of a DDS.

The IRE must be able to provide support for the functional responsibilities of a model administrator (Figure 4). Thus, at a minimum, it must inventory existing organizational models providing basic documentation such as model name, type, version, developer(s), software program(s) required, etc. The IRE must also reflect the interaction between models and other resources, particu-

larly data. Thus it should be able to show which data elements are inputs to models and which are outputs. Access, security, and integrity information can be controlled via the IRE as well (although this begins to overlap some of the functions of an MMS). In particular, the IRE should provide a uniform way of cataloging the underlying assumptions of a model so that they can be understood in a straightforward manner by people other than the model developers. This is a vital step in making models a sharable resource.

In addition to the valuable cataloging functions which an IRE provides, it must also be able to serve as a decision support system to the MA in establishing modeling strategy and policy within an organization. The MA may, for example, want to analyze model usage within the past year and project usage in upcoming periods. This may reveal a need for development of new models which may, in turn, reveal a need for certain data items not currently in the data inventory. Alternatively, the MA may want to devise a cost/benefit model (via the MMS) using information from the IRE in order to determine the feasibility of a particular modeling project.

The uses of an IRE are as varied as the job responsibilities that are assumed by the Model Administration function. The key point, however, is that an IRE is a vital management tool, not only in model administration but in data administration and overall information resource management as well. The IRE is the central repository of the semantics and structural declarations associated with the data and models. The recognition of models as a corporate resource should contribute to the evolution of a DDS into an IRE.

The other weapon in the model administration arsenal is the model management system. The MMS is to models what the DBMS is to data, i.e. a software system which provides for the creation, manipulation, and access of models. The objectives of an MMS are primarily [5]:
application independence,
multiple views,
DBMS compatibility,
knowledge-based capability.

Application independence implies that an MMS should accommodate broad classes of models, e.g. math programming, regression, simulation, spreadsheet, etc., within an integrated environment. Just as it's possible to define data entities and relationships dynamically in a DBMS environment regardless of data type, so should models be definable independent of model type. Again this requires a uniform means of model representation and model description. An MMS should also provide multiple views of a single model so for example a user can look at a linear programming (LP) model either graphically, algebraically, or as a sparse matrix. This allows the modeler and/or decision-maker to interact with a model at a level most compatible with the user's own view of that model. DBMS compatibility reinforces the recurrent theme in this paper of the close dependence between models and data and the need for structuring model administration in a way similar to data administration. Thus an MMS should be developed based on existing DBMS principles. The knowledge-based capability refers to the encyclopedia concept introduced above which allows the MA to use the IRE as a decision support system for model administration. Knowledge about models is as important as the models themselves in terms of understanding and interpreting what they mean and how they can be used in a decision-making context.

The components of an MMS are shown in Figure 6 and many of them correspond to their DBMS counterparts. Critical to the success of an MMS is a model description language (MDL) which allows users to specify models in a uniform fashion independent of any particular physical implementation. The most logical choice for an MDL is probably an algebraic language similar to GAMS [1] which facilitates the description of a wide range of mathematical models. The key advantage of an MDL is that it allows the user to concentrate on model description without worrying about how the model gets solved. Thus, the modeler should be able to describe an LP problem or a spreadsheet model using the same MDL and let the MMS worry about how to generate the necessary representation for the solution phase.

A model manipulation language (MML) allows the user to perform operations on models such as update, display, solve, and link. An MML may take the form of a query language [Blanning 1984] or it may be embedded in a host language like Fortran or Pascal. The model control system (MCS) is the "nuts and bolts" of the MMS governing access, storage, and retrieval protocol at the physical level of implementation.

MMS Component	DBMS Counterpart
Model Description Language (MDL)	Data Description Language (DDL)
Model Manipulation Language (MML)	Data Manipulation Language (DML)
Model Control System (MCS)	Database Control System (DBCS)
Solution Library	

Fig. 6. Components of an MMS and Their DBMS counterparts.

The MDL, MML, and MCS are all model counterparts of corresponding DBMS components. The solution library, however, is unique to the modeling domain. It consists of a set of algorithms for solving various kinds of models, for example a simplex algorithm for LP models or the Gauss-Seidel technique for simultaneous equation models. The solution library must contain a sufficient portfolio of algorithms to accommodate the particular application environment it will serve. The GXMP system [4], an MMS for math programming (MP), contains Fortran source code algorithms for solving various kinds of MP prob-

lems. It is critical to note that the solution library may include spreadsheet packages as well. In fact, it would not be unreasonable to view the solution library as the existing portfolio of spreadsheet and other modeling packages within an organization (see next section).

The various components of an MMS might interact as depicted in Figure 7. The figure depicts a possible configuration for an MMS and not necessarily a recommended architecture. We will use this architecture for purposes of discussion. Models are built via the MDL which results in model banks and databases containing model
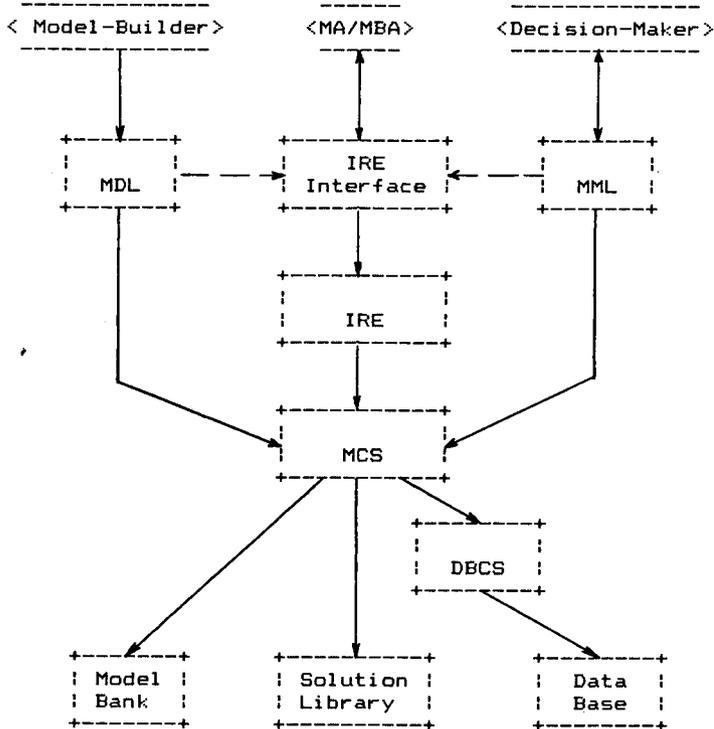

Fig. 7. The Components of an MMS.

equations and corresponding data respectively. The databases may already be part of an organization's existing data resources. Users access the models via the MML either in query or host-embedded mode. If a model is to be solved, the appropriate solution algorithm must be selected by the MCS and the model description then transformed to the representation required by this algorithm. Reporting of model results may then be effected using capabilities available either in the solution package or at the MMS level. The IRE keeps track of the various models, data, and algorithms currently in the system. If the IRE is fully integrated with the MMS, then the MA/MBA can use the MML to access it. Otherwise, the IRE will have its own interface. Notice there are three logically distinct entry points to the system corresponding to the type of user. Model builders use the MDL, model users the MML, and model administrators the IRE interface which may include both the MDL and MML. In general, these three types of users will have different needs and require different MMS functions, although in small organizations, the user roles will likely overlap.

8. A Hypothetical MMS Example

MMS's on the scale that the authors envision are still waiting to be built, at least on the commercial level. Although some commercial packages claim model management capabilities, they are quite rudimentary in most cases and directed primarily at a *single user* environment. None fulfills all the objectives of an MMS listed above, the intent of which is to provide an *organization-wide* model management capability. The following suggests briefly an approach for building an MMS from an existing spreadsheet environment. The assumption is that there is a mix of spreadsheet packages and personal computers operating independently in single user environments.

Figure 8 depicts this situation in the context of a small organization. A local area network (LAN)
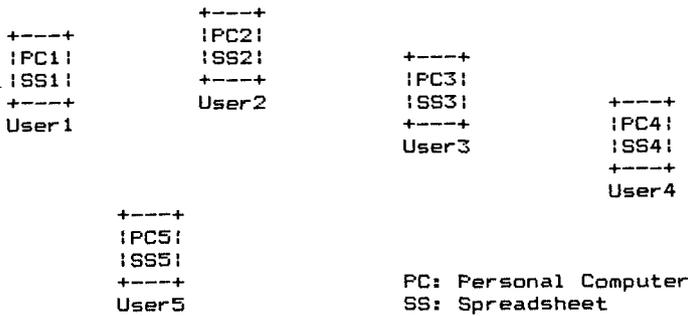
```
                 +---+
+---+            !PC2!
!PC1!            !SS2!            +---+
.SS1!            +---+            !PC3!
+---+            User2           !SS3!            +---+
User1                            +---+            !PC4!
                                 User3            !SS4!
                                                  +---+
                                                  User4
          +---+
          !PC5!
          !SS5!
          +---+            PC: Personal Computer
          User5           SS: Spreadsheet
```

Fig. 8a. Single User PC/Spreadsheet Environment.

```
                              +---+
                              !PC5!----
                   +---+      !   !
                   !PC4 !     +---+
                   !   !      User5
                   +---+              +---------+
+---+              User4      +-------+  IRE DB !
!PC3!                                 !  Models !
!   !                Central ! MMS/   !  Data   !
+---+                   PC/   !        ! SS1,SS2,!
User3                Controller! IRE   ! SS3,... !
                     +---+     +--+---+ +---------+
                     !PC2!           !
                     !   !      +--+--+     Disk
                     +---+      !PC1!
                     User2      !   !
                                +---+
                                User1
```
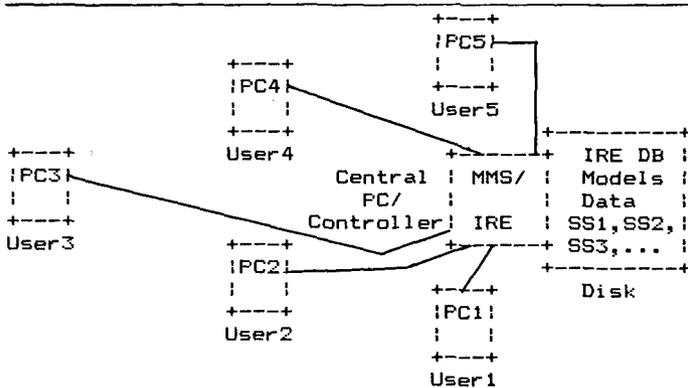
Fig. 8b. An MMS Configuration for Integrating Single User PC/Spreadsheet Environment.

is implemented with a centralized processor and disk storage capacity. The various spreadsheet software resides on the disk as do the IRE, the MMS, and the models themselves. All users interact with the same MDL when describing models. This language ideally would be a superset of existing languages, preferably algebraic in form. When a user wants to analyze an existing model, (s)he specifies the model name and the target spreadsheet. The model description is transformed to the representation required by the target spreadsheet and then downloaded to the individual's PC for custom analysis. When analysis is complete, the model is uploaded to the central CPU and any changes which have been made to the model are catalogued by transforming the new spreadsheet representation back to the MDL and storing it on disk. All model transactions at the centralized level are channeled through the IRE so it contains a current account of the modeling resources.

This is clearly a thumbnail sketch of an MMS configuration which leaves many crucial details unspecified such as how to manage data in this environment and where to get software that performs these MMS/IRE functions. Nevertheless, software packages already exist for spreadsheet-to-spreadsheet transfer and one spreadsheet package currently offers an algebraic language, so many of the necessary ingredients are already available in the marketplace. The distributed computing technology in the form of LANs also exists today so the hardware is available as well. The advantage of this kind of MMS structure is readily apparent. It permits model sharing, standardizes model description, and contributes to a reduction of model redundancy and inconsistency. The distributed environment provides a centralized management capability which allows the organization to begin to control the modeling resource while still maintaining the flexibility and individual productivity which spreadsheets provide.

The above structure is by no means the only approach possible to model management in a spreadsheet environment. One could characterize the LAN as horizontal integration and contrast it to vertical integration wherein micros are tied directly to minicomputers or mainframes which perform the model management functions. Other variations and combinations of these approaches exist as well but they all have in common the integration of the hardware and software resources with the existing organizational inventory. It is inevitable that management will move to control the PC/spreadsheet resource and it is quite likely that this control will take a form similar to that described above.

9. Conclusions

The rapid proliferation of personal computers and spreadsheet software within organizations has exposed the need for effective model management and administration. Parallels to the evolution of data management and data administration are evident when viewed in the context of Nolan's model of evolution of DP activity within organizations. The evolution from proliferation of hardware to managing hardware to recognizing data as a resource and then to data administration in the mainframe era seems to be repeating itself now in the PC era. If this is the case, we may expect to see the current proliferation of micros lead to management strategies for controlling micros which will in turn lead to the recognition of models as a resource requiring organizational model administration.

In this paper, we have discussed the various aspects of model administration including role, function, and structure. Particularly important is the recognition of the technological vs. administrative duality of model administration and the impact on organizations and information software system requirements. An information resource encyclopedia and a model management system emerge as critical software tools for supporting model administration. A horizontal integration strategy is one feasible alternative for implementing model management in a spreadsheet environment, although by no means the only alternative. In summary, although modeling and model management have been longstanding problems in many organizations, it seems that the advent of PCs/spreadsheets may provide the needed impetus for organizations to recognize, control, and integrate the modeling resource.

References

[1] Bisschop and Meerhaus, Toward successful modeling applications in a strategic planning environment, Large Scale Linear Programming, vol. 2, ed. by G.B. Dantzig, M.A.H.

Dempster, and M.J. Kallio, IIASA, Laxenburg, Austria, 1981, 711-745.

[2] Blanning, R.W., Conversing with management information systems in natural language, CACM, 27, 3(March 1984).

[3] Codd, E.F., A relational model for large shared data banks, CACM, 13, 6(June 1970).

[4] Dolk, D.R., A knowledge-based system for LP modeling, Naval Postgraduate School Technical Report NPS-54-83-012, January 1983.

[5] Dolk, D.R. and Konsynski, B.R., Knowledge representation for model management systems, November 1984. IEEE Transactions on Software Engineering.

[6] Fromm, G., Hamilton, W.L., and Hamilton, D.E., Federally Supported Mathematical Models: Survey and Analysis, U.S. Government Printing Office, Washington, D.C., 1974.

[7] Konsynski, B.R., Databases for System Design, Future Directions in Databases, ed. by G. Ariav and J. Clifford, Ablex Press, 1985.

[8] Leong-Hong, B.W. and Plagman, B.K., Data Dictionary/Directory Systems: Administration, Implementation, and Usage, Wiley-Interscience, 1982.

[9] Nolan, R.L., Managing the computer resource: A stage hypothesis, CACM, 16, 7(1973), 399-405.

[10] Nolan, R.L., Managing the crisis in data processing, Harvard Business Review, March/April 1979, 115-126.

[11] Will, H.J., Model management systems, Information Systems and Organization Structure, ed. by Grochla, E. and Syzperski, N., Walter deGruyter, Berlin, 1975, 467-482.