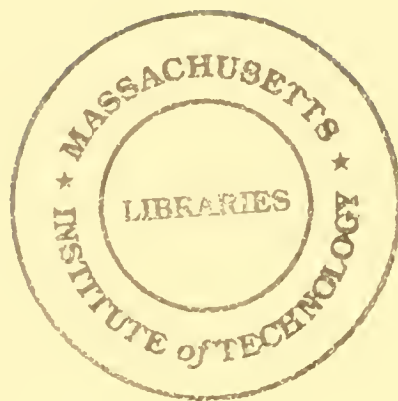


MIT LIBRARIES

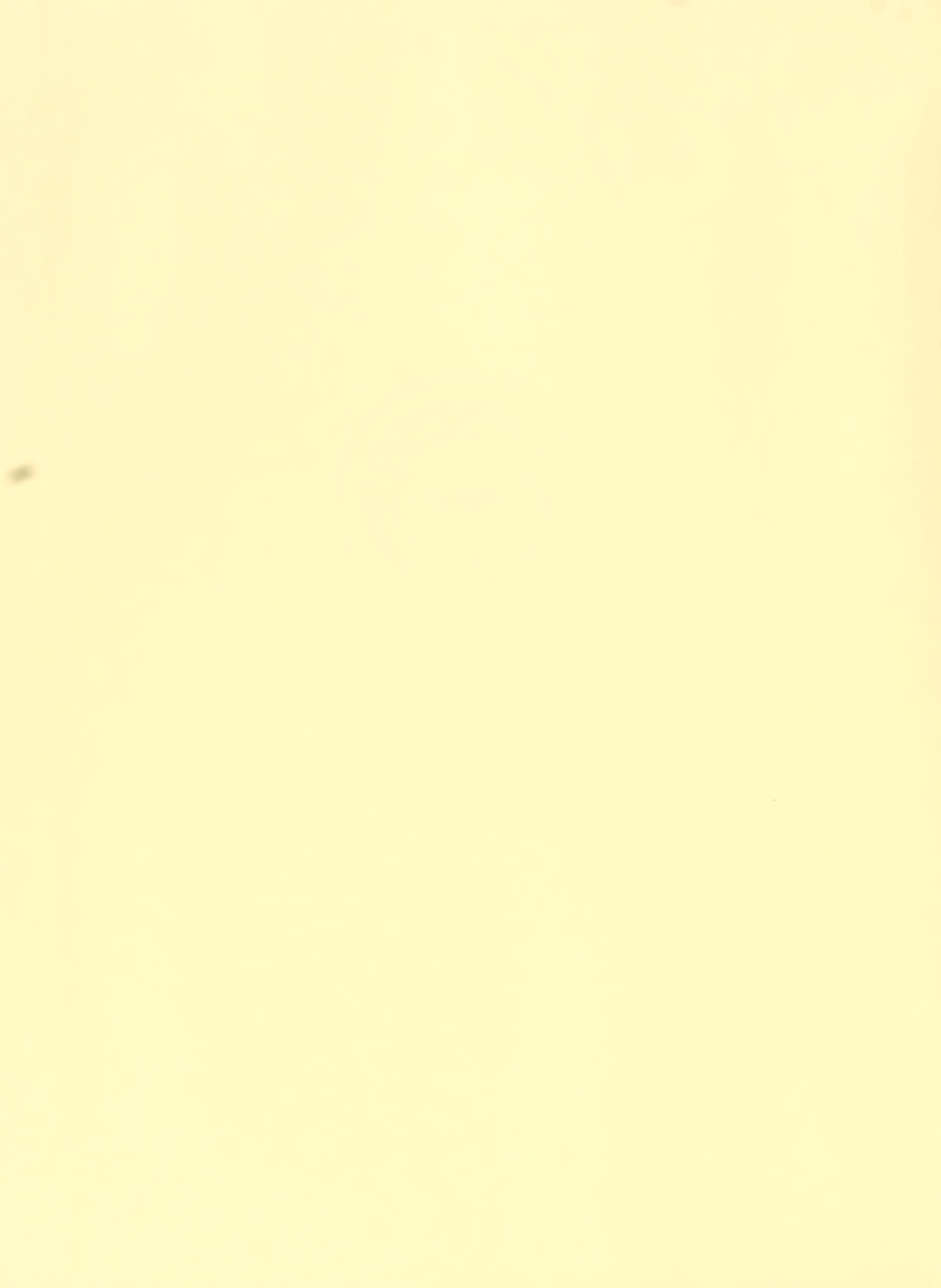


3 9080 02238 6061

BASERENT







HD28  
.M414  
no. 3570-  
93

DEWEY

WORKING PAPER  
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

THE DECISION DATA BASE

Jeremy F. Shapiro

Sloan WP# 3570-93-MSA

June, 1993

MASSACHUSETTS  
INSTITUTE OF TECHNOLOGY  
50 MEMORIAL DRIVE  
CAMBRIDGE, MASSACHUSETTS 02139



THE DECISION DATA BASE

Jeremy F. Shapiro

Sloan WP# 3570-93-MSA

June, 1993

© Copyright 1993, Jeremy F. Shapiro.

MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

NOV 14 2000

LIBRARIES



## THE DECISION DATA BASE

Jeremy F. Shapiro

June, 1993

### Abstract

Advanced Decision Support Systems based on mathematical programming models receive input data from Decision Data Bases (DDB) that are derived from, but different than, corporate data bases. This paper elaborates on the concept of the DDB, its derivation, and its practical significance. Two specific DDB's are examined in detail, one addressing integrated logistics planning and the other addressing dedicated portfolio optimization.



## THE DECISION DATA BASE

James F. Shroyer

June 1973

### Abstract

Advanced Decision Support Systems based on mathematical programming models receive input data from Decision Data Bases (DDDB) that are derived from the different data sources. This paper discusses on the concept of the DDDB in detail, and its practical application. Two specific DDDBs are examined in detail, one addressing inventory control planning and the other addressing detailed portfolio optimization.

# THE DECISION DATA BASE

## Introduction

An increasing number of managers have come to realize that transactional data in their corporate data bases cannot be used directly to analyze many of their important decision problems. The difficulty is especially apparent if the problems involve large sets of numerical data. This is the case, for example, when managers are confronted with facilities location, production scheduling, or other value chain management problems. It is also the case for portfolio selection and related financial decision problems.

Shapiro et al [1993] discuss the application of Advanced Decision Support Systems (DSS's) based on mathematical programming models to value chain management problems. Their development includes a brief discussion of what they call the Decision Data Base (DDB) which is derived from corporate data bases and contains the input data for Advanced DSS's. They note that the DDB has value in its own right in supporting managerial decision making, even if it is not used in the generation and optimization of mathematical programming models.

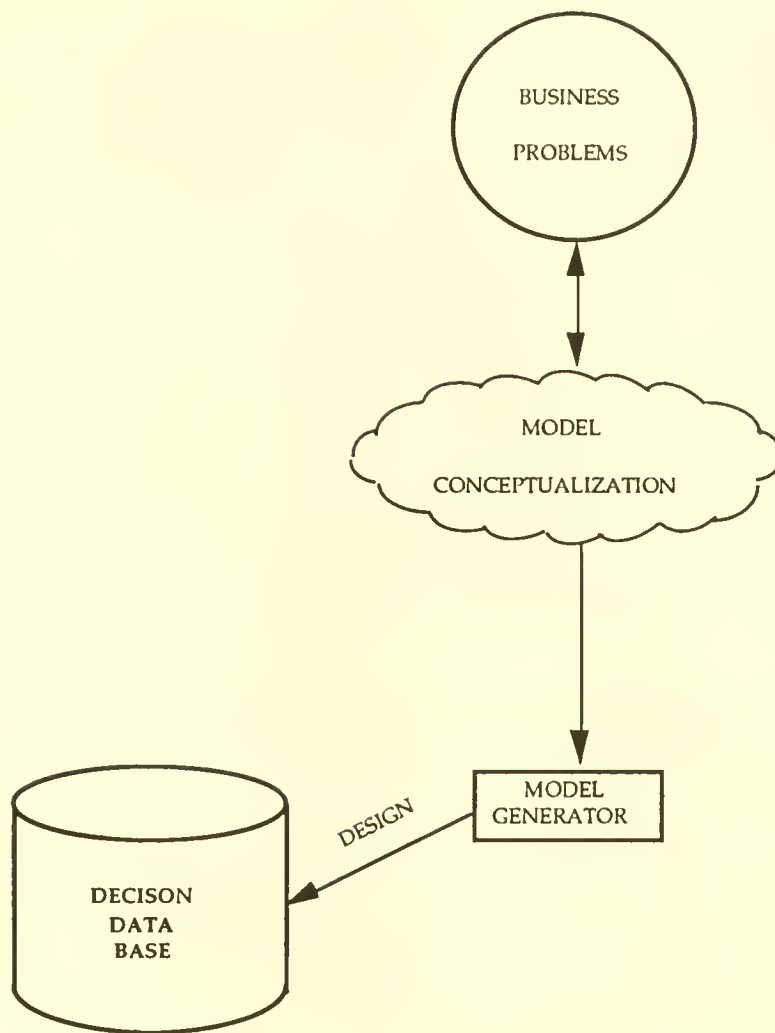
The goal of this paper is to elaborate on the concept of the DDB and on its practical significance. Our development will employ mathematical programming as the primary tool both for analyzing business problems and for suggesting the design and content of DDB's. Thus, the paper also serves to review recent practices in the use of models to support managerial decision making. DDB's could be developed in other ways. However, we have found mathematical programming to be a robust problem solving paradigm that is well suited for these purposes.

## Derivation of the DDB

In contemplating the form and content of any DDB, we require that the collection of business problems of concern be focused to the extent that a coherent family of models can be created to analyze them. In other words, the DDB is the result of inductive business problem analysis using models. This is different than a deductive approach in which the management scientist tries to build a complete model of the company's business which, once completed, will be used to analyze virtually any data intensive decision problem that might arise. The latter approach is doomed to failure because it is impossible to bring to closure.

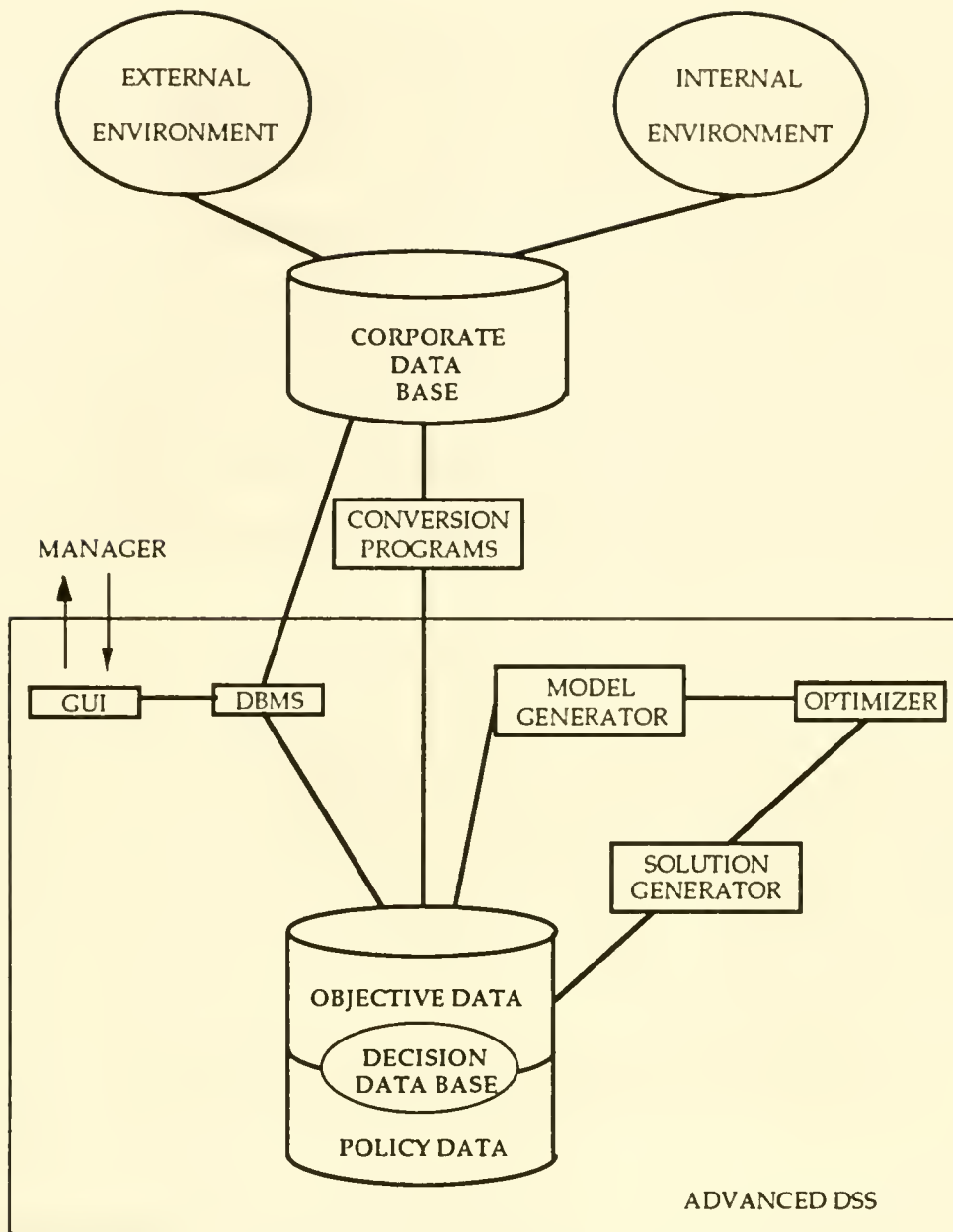
The first step in creating a DDB is depicted in Figure 1. It begins with a domain of business problems to be evaluated. After extensive dialogue with the manager(s) about these problems, the model builder conceptualizes the models that fit the problems. The cloud like graphic description of these steps indicates that the activities of problem articulation and model building are creative and imprecise mental exercises.

It is possible, even likely, that different model builders will derive different models for a given set of problems. In an absolute sense, some formulations may be better than others, and these differences will be reflected in the associated DDB's. However, for the purposes of discussion, we presume henceforth that the suggested family of models accurately and uniquely describes the business problems at hand.



Relationship Between Business Problems and a DDB  
Figure 1

The model generator is a concrete object. It is a software program that realizes the model builder's conceptualization of the necessary models. As shown in Figure 2, the model generator transforms input data into the equations and variables of a mathematical programming model expressed in a form that can be read by a numerical optimizer.



Advanced Decision Support Schema  
Figure 2

At the design stage of an Advanced DSS development project, the design of the model generator determines the structure of the DDB. Implicit in this design is a parsing of the model's data requirements into their natural components which form the basis for the DDB. As shown in Figure 2, we have

divided the DDB into two parts: **Objective Data** that is derived from the company's corporate data base; and **Policy Data** that reflects the manager's judgment about acceptable characteristics of optimal strategies. Objective data might include, for example, costs, capacities, transformation recipes, and transportation activities. Policy data reflects managerial concern about risk, secondary objectives, or constraints on the practical implementation of a strategy. It might stipulate, for example, that no more than 50% of a company's requirements of raw materials can be satisfied by purchases from companies outside the U.S., or, that bond purchases for a fixed income portfolio must be in lots of 1000 bonds. For most applications, the objective portion will be much larger than the policy portion.

Both types of data in the DDB are quantitative, although some of the policy data may be logical or boolean in nature. For example, a constraint that a company's logistics network may contain a distribution center (DC) in Los Angeles or San Francisco, but not both, or a constraint that a dedicated portfolio may contain bonds issued by General Motors or Ford, but not both.

A third possible type of input data omitted in Figure 2 are control parameters for the optimizer. Such parameters are especially needed when the model to be optimized is a mixed integer program. Because the time required to compute an optimal or demonstrably good solution for such a model can be unpredictable. Thus, it is important to impose limits on the extent of computation performed by the optimizer. Moreover, the user may assist the optimizer by conveying his/her judgment about priorities among important decision variables. This type of data could be viewed as policy information about the desired quality and probable structure of decision strategies produced by the Advanced DSS.

Referring again to Figure 2, we note that model generation is data driven at run time. That is, for a given data instance, the model generator checks which components of a decision problem are passed forward from the DDB and generates a model encompassing only those components. In this way, the model generator has the capability to create a variety of models, each tailored to the immediate needs of the decision maker.

The solution generator is a program that takes the output data from the optimizer and parses and organizes it in a manner consistent with the structure and content of the input data fed to the model generator. It uses the internal names of decision variables and constraints selected by the model generator in creating the mathematical programming model. The solution generator interprets and, for some output, suppresses non-managerial technical structures associated with mathematical programming models. Output from the model generator becomes part of the DDB.

We intend and expect that the DDB and the Advanced DSS will reside on a pc or workstation, although the company's corporate data base may reside on a mainframe computer. Figure 2 conveys the idea that an Advanced DSS can be implemented on these platforms using an open architecture approach in which the graphical user interface (GUI) and the data base management system (DBM) can either be purchased off-the-shelf or constructed quickly using a software toolkit. The optimizer can also be purchased off-the-shelf. Only the model generator needs to be handcrafted for the problem solving domain of the Advanced DSS.

The perspective that the optimizer is a "black box" which can be purchased off-the-shelf merits emphasis. It is not widely appreciated that highly effective linear and mixed integer programming packages for desktop computers can be acquired at a modest cost. After more than 40 years of development, the research



community has developed efficient algorithms for these types of mathematical programming models. Coupled with phenomenal strides in the numerical processing capabilities of microprocessors, these packages allow optimization of large scale models in times that are commensurate with those associated with mainframes of less than 10 years ago. Of course, the demand for faster computation and an ability to optimize larger models is never ending. Still, the absolute capabilities of today's desktop computers allow business problems of significant size and complexity to be efficiently modeled and optimized on them.

With these advances, optimization of a properly posed mathematical programming model is a reliable and almost routine task. A more challenging task is to conceptualize and implement the model generator. The bulk of the actual work, however, is to create the DDB. In terms of the time required to implement and validate an Advanced DSS, perhaps 80% or 90% will be spent in developing the data handling routines of the DDB, organizing data, and making validation runs, while only 10% to 20% will be spent on the model generator.

The discussion thus far has assumed that a DDB is defined in terms of a coherent class of business decision problems. In later sections, we discuss specific DDB's, one addressing logistics problems and the other addressing dedicated portfolio selection problems, in order to be more concrete about their construction, meaning and use. We have chosen such disparate applications in order to examine the DDB from radically different perspectives.

### **Business Process Redesign, Integrated Planning and the DDB**

An Advanced DSS is often sought by management to promote integrated or coordinated planning within the firm. Mathematical programming models are well suited to the task of unraveling the complex interactions and ripple effects that make integrated planning difficult and important. For such

applications, the DDB reflects the content and level of data detail that must be communicated among managers with differing functional responsibilities to achieve integrated planning. A specific case is discussed in the following section.

The role of the DDB in integrated planning is complementary to current software developments aimed at promoting business process redesign (e.g., see Davenport [1993]). For example, Winograd and Flores [1987] have proposed a workflow paradigm defined in terms of the interaction between two people conducting business, the "customer" and the "supplier". As goods or services move through the value chain, the customer at a given stage becomes the supplier of a different customer at the next downstream stage of the chain. New software is needed to facilitate formal or informal negotiations about the terms of satisfaction between customer and supplier, and agreement about when these terms have been met.

Advanced DSS's and DDB's for integrated planning are needed to complement business process software and procedures because decisions made by customers and suppliers will tend to be myopic. It is critically important for customers and suppliers, especially if they work within the same firm, to be given guidelines that are effective from a more global, integrated viewpoint. Conversely, the implementation and use of new business process software should greatly facilitate the creation of accurate and timely data for the purposes of higher level, integrated planning by Advanced DSS's.

### **DDB for Integrated Logistics Analysis**

The discussion in this section represents an amalgam of facts and experiences from several integrated logistics modeling studies performed for retail distribution companies. We begin by considering the logistics network of the distribution division of a retailing company operating in Illinois, Wisconsin

and Indiana. A simplified form of the network is displayed in Figure 3. The division handled in-bound transportation, warehousing, inventory management and out-bound transportation for more than 50,000 SKU's shipped to retail stores where the products are sold to the public. Total annual logistics costs exceeded \$100 Million.



Logistics Network of Retail Distribution Company  
Figure 3

The retail stores were comprised of corporate stores owned by the parent company, franchise stores with which the parent had long-term contractual arrangements, independents, and specialty stores in a small company recently acquired by the parent. The total number of stores in all categories was approximately 600. The division operated 7 distribution centers (DC's): A large

DC in Chicago and 6 medium-sized DC's located in or near other cities. The DC's were stocked by approximately 400 suppliers located throughout the U. S. and Canada. Foreign suppliers were considered to be located at the port of entry of their products.

Senior management of the division wished to analyze a range of questions about the structure and operating rules of the logistics network. The time frame for the analysis was one year; study years included the next calendar year and the two calendar years after that. The questions to be answered included

- What is the optimal number and location of DC's?
- Should each DC handle all product lines or should some product lines be handled by specialized DC's?
- Which DC should serve each market?
- Which supplier locations should serve each DC?
- What is the tradeoff between logistics cost and service level as measured by the maximal time to transport products from any DC to any customer?
- What is the additional cost of servicing each customer for all its products from a single DC?

A mixed integer programming model was well suited to study questions such as these. The model was a snapshot or single period model encompassing one year of the company's operations. (See Shapiro [1985] or Williams [1990] for further details about mixed integer programming models and their application to logistics.)

A central data construction for integrated logistics planning models such as the one that was used in this application is the definition of the set of

"products" which are actually product families. In this case, after considerable discussion, we chose 40 products given by

8 warehouse groups × 5 types of customers

Each warehouse group was comprised of products with similar handling and distribution characteristics. These groups had been used for planning and control purposes in the company for several years. The types of customers were corporate, franchise, large independent, small independent, and specialty company stores.

The product definition was chosen so that each type of customer had similar DC assembly and store delivery characteristics. That is, for each warehouse group, the work of receiving, assembling and re-loading orders constituting full truckload shipments out-bound from the DC's varied among, but not within, each of the 5 types of customers. Moreover, the time spent by the delivery truck driver unloading at the stores varied among, but not within, each of the 5 types of customers. Thus, the 40 products reflected a complete spectrum of handling and transportation differences due to differences in physical handling and customer characteristics.

The next set to be defined was the set of customers. This definition followed naturally from the definition of products. The 600 stores were used to define approximately 200 customers as follows. Large stores were treated as separate entities in their given geographical locations. There were approximately 50 of such stores. The remaining 150 customers were aggregations of similar types of smaller customers in close geographical proximity. Each type of customer had positive demand for each of the 8 warehouse groups of products.

The set of suppliers was similarly defined. The largest 40 were treated as separate entities in their given geographical locations. Each of these suppliers

were sources for only a few of the 8 warehouse groups. The remaining suppliers were aggregated into approximately 50 supplier zones.

PRODUCT INDEX SET
CUSTOMER INDEX AND LOCATION SET
SUPPLIER INDEX AND LOCATION SET
DC INDEX AND LOCATION SET
RESOURCE SET
SUPPLIER COSTS AND CAPACITIES
IN-BOUND TRANSPORTATION ARCS: COSTS AND CAPACITIES
SUPPLIER DIRECT TO CUSTOMER ARCS: COSTS AND CAPACITIES
RESOURCE CAPACITIES AT THE DC'S
PRODUCT COSTS AND TRANSFORMATION RECIPES AT THE DC'S
INDIRECT VARIABLE AND FIXED COSTS AT THE DC'S
INTER-DC SHIPMENT ARCS: COSTS AND CAPACITIES
OUT-BOUND TRANSPORTATION ARCS: COSTS AND CAPACITIES
MARKET DEMANDS
POLICY PARAMETERS

Data Files in the DDB  
Integrated Logistics Advanced DSS  
Table 1

These definitions of the sets of products, customers and suppliers, were the prerequisites for developing the numerical, objective data for an integrated logistics model. A listing of the data files in the DDB are given in Table 1. Most of the data in this table are self-explanatory. Resources at the DC's refer, for example, to labor hours or square feet of storage space. They may also refer to

quantities such as throughput of a particular product upon which inventory handling and holding costs are based. Transformation recipes refer to processes whereby input products are transformed to output products; for example, when products are assembled into orders ready to be shipped to the stores.

The in-bound and outbound transportation arcs constituted the biggest data files in the DDB. The costs associated with movements along the inbound arcs were based on industry rates for truckload movements and the distances between suppliers and existing and new DC's. Approximately 10,000 to 20,000 such arcs were created where the actual number depended on the number of new DC sites to be evaluated. Since most deliveries from DC's to customers were by company-owned trucks, regression analysis on historical data was performed to determine rates and costs for the out-bound arcs. Again, depending on the number of new DC sites being considered, the number of arcs linking DC's to customers ranged from 15,000 to 25,000.

Policy constraints for this family of logistics problems included options allowing the decision maker to specify whether customers were to be sole sourced for each product family, or not. A second policy parameter was the allowable service distance between a DC and the customers it served. This parameter was used to delimit the set of permissible out-bound arcs. A third type of policy constraint were multiple-choice constraints on DC locations. For example, a constraint stating that at most one new DC could be selected from a set of three potential DC sites.

The strategic logistics study that led to the creation of the DDB for integrated logistics analyst for the distribution company was successfully carried out. Many scenarios of the distribution company's future were modeled and optimized. From these runs, senior management identified a redesign of their logistics network with indicated savings of several million dollars in total

logistics cost. The redesign involved a reduction in the total number of DC's in operation and a partial specialization of some DC's to handle a limited number of product lines.

After the study was completed, other important uses of the DDB became apparent. One use was to support quarterly and monthly tactical logistics planning. The need for tactical decision support was particularly acute during the period when the network of DC's was being redesigned. A second potential use of the DDB was as a cost control mechanism. By comparing actual DC operating costs and transportation costs against those projected by the DDB and the optimization runs, management could develop exception reports identifying operational inefficiencies.

### **DDB for Dedicated Portfolio Selection**

The discussion in this section is based on our experiences developing analytic engines for dedicated portfolio selection systems based on mathematical programming models. These were data management and modeling systems used by investment banking firms to evaluate the re-structuring of all or part of a dedicated portfolio comprised of government and corporate bonds and other fixed income instruments. Analysis by models was provided as a service of the investment banking firm to pension managers and other fixed income portfolio managers looking to re-structure their portfolios. Although they were mainframe systems, pc versions based on the open architecture schema of Figure 2 would be straightforward to implement.

The model constraints were of two types. First, for each period (usually a month) in the planning horizon (10 to 40 years), there was an asset/liability balance equation:



- + This period's cash flow from coupons and principal repayment of bonds held in the portfolio.
  - + Cash flow from re-investment of cash surplus from the previous period.
  - + Cash flow from borrowing or other sources during this period.
- =
- + Liabilities forecast for this period.
  - + Re-payment of borrowing in the previous period.
  - + This period's cash surplus which will be re-invested.

Note that the model addressed only the question of which bonds to purchase here-and-now to meet future liabilities. The assumption was that the bonds would be held to maturity. In other words, they would not be sold before that time. Thus, the complications of forecasting future bond prices and incorporating future sell and buy decisions in the model were excluded. Similarly, the model incorporated only the one decision choice of 30-day government bills for re-investment of cash surpluses.

The second type of constraint were policy constraints based on attributes of the bonds. They were imposed by portfolio managers as surrogates for risks associated with the dedicated portfolio selection decisions. Attributes included the here-and- now cost of bonds whose cash flows would (more-or-less) cover forecasted liabilities. Other attributes included rating, average age, duration or yield to maturity of the bonds. Computation of attribute data such as these required tailored routines in the conversion programs shown in Figure 2.

The DDB for this application is shown in Table 2. The lot size quantity in the BOND DATA refers to the integer number of bonds in the minimal lot size that could be purchased (e.g., 1, 10, 100). The conditional minimum refers to the

quantity of bonds of a given bond type that must be bought if any bonds of that type are bought at all (e.g., either 0 bonds or at least 1000). Cash flow pairs refers to combinations of periods and cash flow for those periods when cash flow is positive.

BOND SET

PERIODS SET

ATTRIBUTES SET

BOND DATA – For each bond: name; market price; lot size; absolute minimum; conditional minimum; maximum; attributes; cash flow pairs

PERIOD DATA – For each period: liability payments; re-investment rate; minimum and maximum on cash surplus balance; borrowing rate; maximum borrowing

COST FUNCTION AND ATTRIBUTE CONSTRAINT DATA – Cost function; average constraints; only constraints; each constraints; total constraints; logical constraints

Data Files in the DDB  
Dedicated Portfolio Advanced DSS

Table 2

The default objective function of the model was total cost minimization, but any weighted combination of attributes and cost could be selected as an alternative. As shown in Table 2, the model generator used these attributes values for individual bonds to write "average" constraints on the entire portfolio with respect to these attributes. The "only" constraints were used to screen bonds as candidates for the portfolio based on their attributes. The "each" constraints

were used to limit total investment in individual bonds based on their attributes. The "total" constraints were used to limit total investment in specified sets of bonds based on their attributes. Finally, the logical constraints were a variety of boolean constraints on bonds such as "if bond A is selected then bond B cannot be selected," or "at most one of bonds A, B, C may be selected," and so on.

### **Data Aggregation and Other Estimations and Transformations**

As we have seen, data aggregations are both necessary and desirable for DDB's that address tactical and strategic value chain problems. Due to the broad scope of these problems and the corresponding scope of the models for analyzing them, products, customers, suppliers, time periods and other factors must be aggregated if the models are to be of a manageable size. The same or similar data aggregations are also desirable if the manager is to achieve a high level view of his/her problems. For example, when considering global inventory and production schedules for the next quarter, the manager will obtain better insights by reviewing data aggregated into product families that number in the tens rather than SKU's numbered in the thousands or ten thousands. Moreover, sales forecasts for the next quarter should be based on the same or similar aggregations of finished products. For many types of businesses, regional forecasts should also be based on customer aggregations into market zones.

It is well known that traditional accounting data must often be transformed if they are to accurately describe costs in a DDB. For example, allocations of indirect and overhead costs based on historical levels of volume must be taken out of unit cost figures and treated as separate volume and non-volume dependent costs. This is one of the major tenets of activity based costing (ABC) (see Cooper [1988]).

Another difficulty with standard accounting data is that important costs may be bundled together, thereby obscuring decision options available to management. For example, many suppliers are paid for their products delivered to the company's facilities. In-bound transportation costs are imbedded in the amounts paid to these suppliers. In order to decide whether or not to take over certain in-bound transportation activities, the company must determine or estimate the supply costs FOB the suppliers' facilities, and the in-bound transportation costs from these facilities.

By contrast to the discussion above, sometimes aggregate accounting data needs to be refined for the purposes of decision making. For example, in the integrated logistics DDB discussed above, an average unit cost per mile for on-bound transportation was judged to be too inaccurate. Instead, statistical regression methods were used to compute origin-destination and product specific transportation costs.

### **Comparison of Two DDB's and Advanced DSS's**

The data listed in Tables 1 and 2 are very different, reflecting the differences in decision making between logistics planning and portfolio management. However, procedures that were used to design and implement Advanced DSS's for these applications and their associated DDB's had a great deal in common. In this section, we elaborate on the similarities and differences.

#### **1. Availability of Data and Time Required to Assemble a DDB.**

Our experience has been that data for a logistics DDB requires two weeks to three months to assemble. The actual time depends on the size and complexity of the company's operations, the state of the corporate data base, the number of people involved in converting the data, and several other factors. The monolithic depiction of the corporate data base in Figure 2 is misleading because

data will reside in different data bases in most companies. Some data, such as capacities and indirect costs, may be approximate, especially in the first version of a logistics DDB.

Further, as we indicated above, considerable aggregation of the products, markets, and suppliers may be necessary and desirable for effective tactical and strategic logistics planning. The design and implementation of meaningful aggregation procedures may require several weeks if the problems to be addressed are large and complex. Even after aggregation, the files in the DDB pertaining to in-bound and out-bound transportation arcs will be large and require time to generate and verify.

By contrast, dedicated portfolio DDB's can be assembled in just a few days. This is because electronic retrieval of financial data is better organized and more efficient than it is for logistics data. Financial data is usually more accurate due to its intrinsic nature and the exigencies of global trading. However, once the portfolio manager has identified the strategy that he or she wishes to implement as the result of running several scenarios, a final verification of prevailing market prices for the different bonds and other fixed income instruments, and a final optimization run, are usually required.

## 2. Permanency of Data and DDB's.

Much of the data in the logistics DDB is stable in that it changes slowly over time. Costs and capacities may not change significantly over a period of several months. The rate of change of other data depends on whether the problems being addressed are short-term tactical, in which case we would expect that inventory and demand data will change rapidly, or long-term strategic, in which case we would expect that no data will change rapidly. In either case, the DDB provides an effective mechanism for tracking the real world integrated logistics system being analyzed.

On the other hand, critical data in the dedicated portfolio DDB such as bond prices are quite transient. Although changes in bond prices over just a few days may not be large in absolute terms, they are relatively large for the purposes of exact portfolio optimization. A reduction of 0.1% in the cost of rebalancing a portfolio of \$100 Million is \$100,000.

As they are currently used, dedicated portfolio DDB's are viewed as temporary data sets for analyzing how best to rebalance portfolios. The DDB is created, used to generate optimization models over the course of a few days or weeks, and then abandoned after the exercise has been completed. This seems short sighted. An alternative would be to maintain and update the DDB after the rebalancing has been completed and use the updated data, especially the values of the portfolio's attribute constraint coefficients, as a diagnostic for deciding when the portfolio should once more be rebalanced.

### 3. Similarity of Models.

Mixed integer programming models were used for both applications despite the large differences in the real world business decision problems that they were describing. At the purely mathematical system level, the models have considerable similarity. The similarity could be exploited to translate the dedicated portfolio optimization problem into a pseudo-logistics planning problem. In the recast problem statement, each bond type acts as a potential source of cash flow to be supplied to liabilities viewed as sinks with associated cash requirements.

The artificial but accurate recasting of problems that are not logistics problems as pseudo-logistics problems has conflicting implications to the construction of effective DDB's and Advanced DSS's. On the one hand, the artificiality of the pseudo-logistics formulations would interfere with the design and construction of coherent and transparent DDB's. On the other hand, the

ability to recast a wide variety of non-logistics business problems as a pseudo-logistics problems would facilitate the development of a general purpose model generation language, thereby reducing the extent to which the model and solution generators in Figure 2 must be hand crafted. Brown , Northup and Shapiro [1986] report on a such a language developed on this principle. The approach remains an open area of investigation.

#### 4. Realism of Models.

The usefulness of the DDB's for the two applications is clearly related to the realism of the underlying models. In our opinion, mixed integer programming models provide a very realistic description of decision problems associated with integrated logistics planning. The models capture costs, transformation activities, capacities, inventory management, product movements, and facilities location decisions in a manner that is complete and consistent with managerial intuition.

By contrast, the mixed integer programming models for dedicated portfolio optimization are a less realistic fit to the problems they are supposed to analyze. The main reason for the lack of realism is their deterministic view of the future. The models do not directly address the primary responsibility and concern of financial managers; namely, to control risk while guaranteeing a superior or acceptable return. As we discussed, the attributes constraints are intended to provide these managers with indirect means for controlling uncertainty, but their effectiveness leaves much to be desired.

For fixed income portfolios, the most important uncertain parameters are future interest rates. These rates not only influence the value of cash surpluses generated in the future, but also future cash flows associated with callable assets such as mortgage backed securities. Extension of the mixed integer programming models to stochastic programming with recourse models provide

much more realistic descriptions of dedicated portfolio problems, in large part because they offer the possibility of incorporating models from finance theory describing interest rate movements and options pricing (see Hiller, Klaassen and Shapiro [1991] or Zipkin [1992] for a discussion of stochastic programming models applied to dedicated portfolio optimization). However, the extensions require considerable more research and development because their form and size makes them difficult to manage and implement.

By their very nature, stochastic programming models of dedicated portfolio optimization problems would require the creation of extremely large DDB's. In effect, the DDB would need to contain data describing each of a very large number of scenarios of an uncertain future. Note also that the box in Figure 2 labeled "Conversions Programs" would contain complex and sophisticated interest rate forecasting and other descriptive models from finance theory.

Of course, probabilistic analysis of integrated logistics planning problems may also be desirable. In many instances, this can be accomplished by running deterministic models under different scenarios of an uncertain future to see how optimal strategies vary. Each scenario might be determined by modifying a base case scenario in the DDB.

The stochastic programming with recourse approach discussed for dedicated portfolio problems is also relevant to integrated logistics planning (e.g., see Wagner [1969] or Bienstock and Shapiro [1985]). In effect, a stochastic programming model would simultaneously determine optimal contingency plans for each scenario and a here-and-now strategy that optimally hedges against these plans. Although the approach is technically feasible, it is not yet practical to pursue because the simpler deterministic models are not yet sufficiently understood or accepted.



## Conclusions

Our purposes in this paper were to introduce the notion of the DDB and to demonstrate its importance in decision support. We presented specific illustrations of DDB's constructed for integrated logistics planning and dedicated portfolio optimization, and compared and contrasted their functionalities and uses. We conclude the paper with brief discussions of important related topics not previously covered.

Although the discussion focused on models and systems for decision support, the DSS practitioner must never lose sight of the critical need for accurate data in the DDB. To this end, the "Conversion Programs" in Figure 2 may include a collection of complex descriptive models which, in some instances, may be the most difficult and time consuming task of an Advanced DSS implementation. In addition, these programs should incorporate procedures for error checking. Recent advances in data quality management software are also relevant to the creation of accurate DDB's.

The specific DDB's discussed above related to tactical and strategic planning. A number of new issues arise when one considers DDB's for operational decision support. One is the need to examine decision problems in much greater detail at the operational level than is necessary and desirable for tactical and strategic applications. The form and content of DDB's for operational environments is an area of current investigation.

Finally, we remark that a fundamental assumption underlying the construction of a DDB is that it is defined by the requirement to analyze a coherent class of decision problems. In a large company, we would expect to find multiple classes of problems each with its implied DDB. This suggests a higher level organization of DDB's to support integrated planning at a higher

level. Of particular importance is integrated inter-temporal planning of strategic, tactical and operational decisions. The concept of **hierarchical planning** is relevant to the design of DDB's for this purpose (see Hax and Meal [1975]). Mathematical programming models and methods for hierarchical planning are particularly attractive approaches for creating related and overlapping DDB's (see Graves [1982]).

## References

- D. Bienstock and J. Shapiro [1988], "Optimizing Resource Acquisition Decisions by Stochastic Programming," (with Daniel Bienstock), Management Science, 34, pp. 215-229.
- R. W. Brown, W. D. Northup and J. F. Shapiro [1986], "LOGS: A Modeling and Optimization System for Business Planning," in Computer Methods to Assist Decision Making, edited by G. Mitra.
- R. Cooper [1988], "The Rise of Activity-Based Costing – Part One: What is an Activity-Based Cost System?," J. of Cost Management, (Summer), pp 45-54.
- T. H. Davenport [1993], Process Innovation: Reengineering Work through Information Technology, Harvard Business School Press.
- S. C. Graves [1982], "Using Lagrangean Techniques to Solve Hierarchical Production Planning Problems," Management Science, 28, pp 260-275.
- A. C. Hax and H. C. Meal [1975], "Hierarchical Integration of Production Planning and Scheduling," pp 53-69 in North-Holland/TIMS Studies in the Management Sciences, Vol. 1, Logistics, North-Holland and American Elsevier.
- R. S. Hiller, P. Klaassen and J. F. Shapiro [1991], "A Stochastic Programming Model for Risk Controlled Bond Portfolio Dedication," Working Paper IFSRC No. 183-91, International Financial Services Research Center, MIT Sloan School of Management. (to appear in Proceedings of Quantitative Methods, Super Computers and AI in Finance, London, February, 1992.)
- J. F. Shapiro [1985], "Quantitative Methods in Distribution," Chapter 15 in The Distribution Handbook, edited by J. F. Robeson and R. G. House, Free Press-MacMillan.
- J. F. Shapiro, V. M. Singhal, S. N. Wagner [1993], "Optimizing the Value Chain," Interfaces, 23, pp 102-117.

H, M, Wagner [1969], Principles of Operations Research, First Edition, Prentice-Hall.

H. P. Williams [1990], Model Building in Mathematical Programming, Wiley.

T. A. Winograd and F. Flores [1986], Understanding Computers and Cognition: A New Foundation for Design, Ablex Publishing Co.

P. Zipkin [1992], "The Structure of Structured Bond Portfolio Models," Operations Research, 40, S157-S169.







Date Due <sup>MAR</sup> 2001'

Lib-26-67



MIT LIBRARIES

DUPL



3 9080 02238 6061

