Caveat lector!

This is an early draft of:

- "The Evolution of Decision Support Systems in Consumer Goods Marketing", that appeared as ...
- Chapter 4 of the *The Marketing Information Revolution*

Since electronic versions of this 1994 publication are hard to come by, this 1993 draft is provided as a service to researchers.

If you like the general direction of this writing, please obtain a copy of the book from a library, or purchase a copy online.

This draft is not a definitive work that should be cited.

David Ing
http://coevolving.com

Filename: EVOLMDSS.TXT (WP 5.1) Version: March 29, 1993

The Marketing Information Revolution (Chapter n)

The Evolution of Decision Support Systems in Consumer Goods Marketing

by:

David Ing[1]

The transactional efficiency of delivering goods to the consumer has been aided by the falling cost of computing through the 1980s. PC-based Point-of-Sale (POS) systems have become a standard for retailers, enabling improved customer service through speedier checkout times and better inventory estimates in the stores. Manufacturers routinely use hand-held terminals to record shipments made to distribution centers, or directly onto the retail floor. Operational-level systems can mechanically perform tasks such as replenishment, (i.e. automated item-level reordering), but often do not fully capture the dynamics of today's changing markets. At the tactical level, retail buyers and manufacturers' brand managers struggle to distill macro trends from the multitude of marketing activities which simultaneously occur. The frequency and number of promotions has skyrocketed; competitive positions change with the rapid introduction and withdrawal of new products; consumer segments are constantly being redefined and remapped. "Overnight ratings", "ad hoc analysis", and the "zero-sum game" for market share set the stage for "instant answers" and compression of the planning cycle. In this environment, Marketing Decision Support Systems (MDSSs) need to be more than the eyes and ears of the marketplace; they must provide the marketing intelligence on which future actions may be based.

The capability for "conversational" dialogues with 1970s time-sharing mainframe systems created a vision that business professionals, without the assistance of a programmer, might be able to view and manipulate corporate data through the use of a Decision Support System (DSS). This chapter will review some early concepts of MDSSs (**Marketing** Decision Support Systems), first with a view inside the "black box", and then from a "support system" perspective, contrasted to other interactive computer systems. Some examples of MDSSs commercially available today are then reviewed in

[1]David Ing is an Industry Specialist for IBM Canada Limited, a Technical Advisor to the C2MIT (Canadian Centre for Marketing Information Technologies), and a Lecturer of Marketing at the Faculty of Management at the University of Toronto.

these contexts. Key information technologies which comprise MDSSs are then discussed, followed by some ideas on the roles of managers on the development of these systems within the organization.

1. Concepts: What is Inside an MDSS?

Little, an early (1979) proponent of Marketing Decision Support Systems, defined one as a

> . . . coordinated collection of data, systems, tools and techniques with support software and hardware by which an organization gathers and interprets relevant information from business and environment and turns it into a basis for marketing action[2].

To learn about the business environment, the marketing manager would interact with a "black box" which would have four components:

Figure: A Black Box Definition of MDSSs (Little, 1979)

Data Bank: This would encompass quantitative marketing variables (e.g. sales, advertising, promotion and price) from internal sources, as well as competitive performance and news reports from external sources. "Soft" information from newspapers and from members of the organization would also be included. Marketers might not be satisfied by existing accounting systems, which are oriented toward processing transactions.

Models: These are the manager's conceptions of "how the world works". Using the numerical data stored in the data bank, various hypotheses could be tested, and developed into explicit computational representations to aid planning and decision-making.

Statistics: This is the process of relating models to data. Although sophistical mathematical techniques are included, the most frequent operations are simple techniques such as grouping, aggregation, taking ratios, picking exceptions or tabulating summaries.

[2]John D. C. Little, "Decision Support Systems for Marketing Managers", Journal of Marketing, Vol. 43, Summer 1979, p 11.

Optimization: With an objective of improving the performance of the organization, techniques could range from simple operations (e.g. ranking alternatives), to the more formal (e.g. operations research).

As an alternative to the meetings, studies and reports where marketers exchange ideas on the data and methodologies, an interactive **Q/A** process with the MDSS would permit an electronic method of communicating about analyses[3].

When Little's article was published, marketing was still perceived very much as an art. Little saw an intermediary problem-solving role for the **marketing scientist,** to make sense of the numerous data and measures available. Trained in marketing, statistical methods and database systems, these intermediaries would create quantitative models which could be exercised by novices. Market status reporting (e.g. what is our market share?) would be transformed into market response reporting (e.g. are our promotions as effective as expected?) For the 1980s, Little predicted an increase in magnitude in the amount of marketing data used (i.e. the "POS Data Explosion"), matched by a similar increase in computer power available for analysis. Unfortunately, a shortage of marketing scientists (which, arguably, has continued into the 1990s), was also predicted, due to the scarcity of individuals developing this mix of skills.

As a "black box" definition, Little outlines some ideas of what is inside the MDSS, but doesn't describe the working details of the software components in action. The average marketer can enter questions and receive answers, but probably does not understand how the data, models, statistics and optimization function together to produce the result. In this context, the contents of the black box are solely the responsibility of of the marketing scientist, who embodies his skills and expertise into software. This view reflects the programming tools of the 1970s, of fourth-generation and procedural languages, but such a view would not be inconsistent with approaches based in knowledge-based systems. The revolution in information technology which has occurred over the past decade should not lead us, however, to constrain our definition of MDSSs to the technology available, but to look to the requirements of the marketer. To expand our understanding of an MDSS to a more abstract level, the nature of the marketing questions to be answered and the role of the user in the human/computer conversation is discussed in the next section.

2. Concepts: How Does a DSS "Support" Decisions?

[3]Abridged from Little (1979), pp. 9-11.

Traditional marketing reporting systems often present choices for information much like meals are presented on a restaurant menu. By selecting a choice, the marketer can request a recipe which reproduces a predictable result. Unfortunately, alternatives not appearing on the menu can not be requested, even if the ingredients are readily available in the kitchen. Some marketers, faced with a limited selection, may be prepared to move to the buffet table, where they can take a personal hand in creating their own results. Reporting systems follow the metaphor of step-by-step recipes. Decision support systems, however, enable the user a flexibility to choose ingredients and combine them according to the needs of the moment. The judgement of the marketer can contribute to the final result, and experience is gained with additional iterations through the process. Little's definition of an MDSS does not reflect the dynamism suggested in this metaphor: the final result obtained through the Q/A process is the same, whether it is generated by a novice, or by a master.

To satisfy the requirements for a number of classes of users, Keen suggested a "Modest Proposal"[4] for the generic design of DSS software. The system would be accessible to the novice or executive/casual user (e.g. a manager who only runs pre-written programs); and to the expert (e.g. a marketing analyst able to customize existing programs to solve problems); as well as to the programmer (e.g. a technical professional able to produce efficient code, and "bulletproof" against user errors). In the era of character-based interfaces which prevailed at that time, these requirements presented a tall order. The APL[5] computer language, however, demonstrated a number of features desirable for DSSs: an interactive operating environment, and an ability to tailor "functions" which seamlessly extend operations which already exist in the language[6]. An APL programmer can create conversational

[4]Keen suggested that Natural Language Processing would be the ultimate user interface, but this branch of artificial intelligence was (and still is) under-developed. See Peter G. W. Keen, "Interactive Computer Systems for Managers: A Modest Proposal", Sloan Management Review, Fall 1976, pp. 1-17.

[5]APL was invented as a mathematical shorthand by Kenneth Iverson while at Harvard, and implemented as an interactive programming language on IBM time-sharing systems in the 1960s. The language is still popular, where its strengths in rapid application development and the manipulation of multidimensional arrays are needed, although most novices are likely to prefer the more visual representations of data, as in spreadsheets. See Doris Appleby, "APL", Byte, December 1991, pp. 141-146; a special edition on the 25th anniversary of APL, IBM Systems Journal, Volume 3, #4, 1991; and Kenneth E. Iverson, A Programming Language, New York: John Wiley and Sons, 1962.

[6]In contrast, fourth generation languages (4GLs) typically have functions and a syntax which is different from the compiled language in which they are

routines which accepts fixed values (e.g. 1, 2, 3) from the novice, as readily as the name of a personal program (eg. MYPROGRAM) which returns some values, as created by the expert. Providing the features of portability (i.e. the same user interface on multiple hardware platforms) and a standard library of routines, APL was an early "end user programming" alternative to procedural languages such as FORTRAN or COBOL.

As interactive, online systems became more common, Keen refined the definition of DSSs to differentiate their use from simpler query and reporting applications.

> The label "Support System" is meaningful only in situations where the "final" system must emerge through an adaptive process of design and usage. This process may be needed for a variety of reasons:
>
> **"Semi-Structured" Tasks:** The designer or user cannot provide functional specifications, or is unwilling to do so. [Structured tasks can be automated or routinized, thus replacing judgement, while unstructured ones entirely involve judgement and defy computerization. Semi-structured tasks permit a synthesis of human judgement and the computer's capabilities].
>
> **Adaptive Design:** Users do not know what they want, and the designers do not understand what they need or can accept. [Traditional systems development projects follow a life cycle with end user requirements clearly defined through analysis and specification phases, in advance of any application coding. A "final" system is released to end users only after exhaustive testing. In contrast, a DSS follows a "middle-out" process, where an initial system is implemented quickly, and then gradually firmed-up, modified and evolved].
>
> **Conceptual Evolution:** User's concepts of the task or decision situation will be shaped by the DSS. The system stimulates user learning and new insights, which in turn stimulate new uses and the need for new functions in the system. The difficulty in pre-specifying how a DSS will be used is reflected through this learning process, as its structure evolves in response.
>
> **Structural Evolution:** Intended users of the system have sufficient autonomy to handle the task in a variety of ways, or

built. As an example, linked lists (as pointers) are a basic feature the C programming language, but a 4GL written in C is unlikely to provide such a construct.

> differ in the way they think to a degree than prevents standardization. [Since there is no "standard" or "right" method to reach a decision, the computer interface must support personalization, and a flexibility for the user to shape the system][7].

Transactional reporting systems often provide information similar to that required by marketers, but they differ in the degree of structure. Accounting, as an example, is based on relatively clear structure for classification, e.g. assets as cash, inventory or plant and equipment. Marketing structures, on the other hand, may be based on overlapping and shifting views of the marketplace. Products may be classified fuzzy definitions such as product characteristics (e.g. regular or family size), by category (e.g. gourmet food), or price ranges (e.g. budget or premium). Consumers may be segmented by demographic characteristics (e.g. senior citizens), lifestyles (e.g. baby boomers) or geographic location. Since it is impossible (and impractical) to reduce the data into every conceivable cluster imagininable, marketers are often provided with detailed, disaggregated data, to be reaggregated when needed.

The volumes of data now available make the creation of aggregates and summaries unmanageable without the assistance of I/S (Information Systems) professionals. Unfortunately, the marketing process compounds the above vagaries of classification with trade-offs between conflicting objectives (e.g. market share and profitability), so that it is often difficult to specify exact information requirements in advance. This "semi-structured" decision-making environment does not suit the traditional methodology of project management for software development. The shorter (weekly) planning cycle associated with marketing data is too short to create a project plan, or to ensure the quality of execution. In addition, the resources of a programmer and/or the computer are not best-utilized by creating a multitude of "ad hoc reports" that may be irrelevant in the following week. The role that the traditional I/S organization can play with decision support is limited: they can install the software and provide basic technical assistance, but it is the marketing scientists who must combine marketing and computer skills to create a true MDSS.

Little's definition of an MDSS is not incompatible with Keen's view of a DSS, except in one area: the degree to which the "black box" is "closed". Keen "adaptive process of design and usage" would suggests that an MDSS initially structured by a marketing scientist should be sufficiently "open" such that its shape could be adjusted to personal tastes and/or needs. Beyond

[7]Abridged/adapted from Peter G. W. Keen, "Decision Support Systems: A Research Perspective", Centre for Information Systems Research #54, Sloan Working Paper #1117-80, March 1980, pp. 6-11.

merely adjusting some input parameters through a Q/A interface, the marketer should at least have some flexibility to make some cosmetic changes (e.g. changing decimal places) and perform simple logic modifications at will. Although a marketing scientist is valuable in implementing the more technical aspects of the MDSS, introducing him/her as an intermediary can intrude on the marketer's conceptual evolution during the adaptive design process. At a high level, Little's vision of a Marketing Decision Support Systems is appealing, but its "black box" structure is much an artifact of menu-based, compiled programs in the late 1970s. Marketers today are much more comfortable with the concept of "personal computing", and are skeptical of computer programs which they do not understand. An ideal marketing decision support systems would provide a blend of the black box's internal structure with a flexibility for structural evolution.

3. MDSSs in Practice -- Two Examples

To bring the definitions of MDSSs to life, two very different approaches to satisfying the needs of marketing managers will be described in this section. The first, CoverStory, was designed specifically to meet an information need of consumer packaged goods marketers, and more closely follows the spirit of Little's vision for an MDSS. The second, the Metaphor Data Interpretation System, is used by marketers across many industry segments, and stands truer to Keen's vision. Each example is compared to the definitions of MDSS discussed above, and their strengths and limitations are highlighted.

CoverStory[8] was conceived as a solution to the "scanner data explosion" in grocery products, where marketers accustomed to brand-level regional aggregates are overloaded with a multitude of marketing measures at the UPC-level by geographic market. The increased volume of data available for analysis has not been matched a similar increase in the manpower to review it; in fact, most marketing staffs have been downsized since the late 1980s. The challenge to "summarize what is important in this data" was answered by an approach which automates the creation of a summary memorandum describing key events in the database.

The procedure follows four steps. Firstly, **marketing models** quantifying the impact of marketing variables (i.e. distribution, price, display, features and price cuts) are created. Products and markets are then **aggregated** into clusters, and ranked to draw attention to the "top few" of

[8]John D. Schmitz, Gordon D. Armstrong and John D. C. Little, "CoverStory -- Automated News Finding in Marketing", Interfaces, Volume 20, #6, November-December 1990, pp. 29-38.

interest, by share or volume change. The most noteworthy products and markets are **decomposed** for further analysis, and the top associated factor changes are scored and ranked. Finally, an English-language **presentation** constructed with sentence templates is filled out and published through a word-processing package.

The CoverStory module is but one part of a larger MDSS, but it reflects a general approach to reducing the large volume of scanner data. Following Little's definition of an MDSS, CoverStory greatest strengths are its model and optimization components. Marketing models quantify an expected outcome from marketing activities, by which actual results may be benchmarked. Optimization is a simple ranking of the variances from expectations, drawing the most significant to the immediate attention of the reader.

Completing Little's definition, the data bank of UPC-level scanner data is obvious in its volume, as are the summary statistics of clustered product-markets. The sophistication of the CoverStory procedure, however, removes the Q/A flavour of an interactive conversation, with written reports instead generated in a batch. Although it is practical to deliver the reports electronically, on a computer screen, the practice of printing hard copy reports may have its root causes in two reasons. The first might be technical: since the data are not updated in real-time, running all product-markets as a single overnight "batch" is less resource-intensive than computing the many online requests for each product-market, one at a time, while the marketer waits. The second is based on the way in which marketers have become accustomed to looking at data: even if brand managers were able to view an electronic form of the same report more rapidly on a computer screen, many would still prefer to print the report on paper, and then physically file it away.

The orientation of CoverStory towards reporting, and away from a "purer" decision support system, is reflected in Keen's definition. CoverStory routinizes the "semi-structured" task of reviewing marketing data into a standardized presentation. Adaptation of the design for other purposes would require that advanced end users be able to pull out specific sub-modules to be combined with alternative procedures. Further structural evolution of this system, however, is likely beyond the capabilities of an end-user, and calls for the skills of a professional programmer. CoverStory emphasizes its primary strength in supporting the conceptual evolution of the marketer's decision environment, while de-emphasizing the other functions of the MDSS. By introducing a "top-down", "exception-based" methodology to reviewing scanner data, the attention of the marketer is focused where his/her their decisions will have the greatest impact.

In contrast to CoverStory's narrowly-defined problem domain,

Metaphor Data Interpretation System (DIS) is a platform which is more consistent with Keen's definition. In comparison to some PC software which have appended graphical user interfaces onto older character-based products, DIS was designed as an object-based interface which would simplify workgroup access to large-scale databases, and manipulation of the data. Relational tables are graphically depicted through a Workstation Tools Data Dictionary, assisting marketers in the creation of complex queries to one or many databases. A Tool-to-Tool Communications facility transfers data from one tool (e.g. a Query to a mainframe database) to another (e.g. a Plot) through a few clicks of the mouse, and Application Capsules may be assembled by connecting the tools together, as arrows between icons. DIS has been used not only by brand managers for the analysis of scanner data, but also by retailers for internal point-of-sale and inventory movements, and by insurance companies for targeting clients.

Figure: DIS Application Capsule

Unlike the "application-orientation" common in most computer environments to date, DIS is architected as programs and data modularized into "objects", providing an environment where applications are developed by linking icons together[9]. Consistent with Keen's definition, DIS can support a semi-structured task/solution process by offering a large set of tools which appears as icons to be connected together as the user works through a problem. As an interactive system, development can follow an adaptive design process. Options within each tool can be selected or modified, and data may be directed in which ever way the icons are visually connected. Conceptual evolution can occur as the marketer incrementally adding tools to manipulate the data as needed. Finally, programs are normally not "compiled" or "locked", so that end users are encouraged to evolve the structure of applications.

DIS is a popular MDSS amongst consumer good manufacturers. A. C. Nielsen offers a Databank[10] service for selected consumer goods manufacturers, providing remote access to over 100 gigabytes of data in 500 databases, with tailored groupings of products and markets for each company. The system is often used for the promotional analysis of events with the consumer (e.g. coupons) or the trade (e.g. allowances and

[9]DIS is described as an object-based environment, as it demonstrates many of the features of object-oriented technology. For a easy description of this area, refer to David A. Taylor, Object-Oriented Technology: A Manager's Guide, Addison-Wesley, 1990.

[10]Personal communication with Laura Reeves, Metaphor Computer Systems, Chicago.

incentives), and for competitive analysis. Manufacturers providing direct store delivery (DSD) can review their sales volumes, and analyze distribution patterns of product from the warehouse to the stores[11]. Retailers have adopted DIS for category analysis and assortment planning, to ensure that the sizes, flavours or colors demanded by consumers are being purchased and stocked. Vendor performance on delivery and profitability are tracked. Recently, there has been interest in using DIS to generate customer profiles, and target offer to segments of the customer base[12].

DIS has the potential to fulfill Little's definition of an MDSS, but the components which he describes are not all immediately "ready-to-use" when the system is delivered. Simple statistics and optimization procedures are easy to build in the DIS environment, and an interactive Q/A process is native. In situations where a marketing data bank does not already exist, however, one must be designed and implemented, often based on the summarized accounting and transactional records[13]. The greatest variability in DIS implementations, however, is the sophistication of models available to the marketer. In some companies, where a backlog of requests to the MIS department has left marketers starved for data, DIS becomes merely a substitute reporting system which has relieves the programming bottleneck. In these cases, marketers can become overwhelmed by a new volume of reports merely relating "what happened", instead of following models which explain "how the world works".

When a company has not had prior experience with marketing models, the semi-structured nature of MDSSs can make the initial specification of an application suite on DIS onerous. A project to develop models for proprietary internal data sources requires special focus to ensure that applications are constructed and readily available for marketers. One approach to introduce a "Marketing Toolkit" first created a "blueprint" of the application suite needed by marketers, and then constructed "proof-of-concept" applications in DIS used to demonstrate the "look-and-feel" and purpose of marketing models[14]". While constructing the full "Marketing Toolkit" was expected to

[11]These applications are discussed in more detail a "Value Assessment Study of the Consumer Packaged Goods Industry" conducted for Metaphor Computer Systems.
[12]Many of these themes are discussed in a "Value Assessment Study of the Retail Industry", conducted by Ernst and Young for Metaphor Computer Systems.
[13]Databases to support tactical decision-making are designed differently from those optimized for transactions. See Willian H. Inmon, "The Atomic Database: Building the Perfect Beast", Enterprise Systems Journal, (date)?, pp. 62-87.
[14]David Ing and Ray R. Serpkenci, "Designing a Retail Marketing Decision-Support Toolkit", presented at the TIMS Marketing Science Conference, March

take some time, the training to raise the level of organizational learning about marketing models could be conducted in parallel.

The above two examples illustrate the current state-of-the-art in MDSSs. A balance must be established between structure, as demonstrated with the rich models built into CoverStory, and flexibility, as demonstrated in the assembly of tools in DIS. This is achieved through the appropriate design in suitable information technologies, as well as management commitment to a vision of developing MDSSs. The next section addresses the first issue by discussing the emerging information technologies on which MDSSs will be based in the near future. The chapter concludes with a discussion of the managerial role in the development of MDSSs.

4. MDSSs in Practice -- Key Information Technologies

The combination of Little's definition with that of Keen creates three basic requirements for information technologies of an MDSS: a wide variety of data should be readily accessible to the marketer; the exchange of ideas, data and methodologies should be supported between marketers through the system; and the system and its shape should be responsive to the user. Three technologies are key in providing an appropriate environment for decision-making: **data repositories** which retain and share news on the marketing world; **workgroup networks** which support the exchange of information between and within marketing teams; and **user interfaces** which can electronically extend the problem-solving capabilities of marketers. The combination of these three technologies provide the foundation for an corporate MDSS platform.

Data repositories were simpler to maintain in the 1970s era of mainframe computing, when all access and technical support was provided centrally on a few large machines. Personal computing and decentralized processing has resulted in data being stored in many places, and in many different structures. One symptom of poor data management is the downloading of data to personal spreadsheets for reformatting and manipulation, not only by single individuals, but by entire workgroups. The solution to this unproductive activity is not to increase the speed at which personal computers can reformat data, but to provide improve access methods so that the data can arrives at the marketer's desk in a more manageable form.

The current standard for multi-user marketing databases is the relational model[15]. Although relational databases are often adopted first for

12, 1993.

[15]IBM DB2, and Oracle are two examples of popular products.

their benefits in improving programmer productivity, marketers can benefit greatly from the flexibility through which data can be selected. Instead of having data structure designed for one specific application (e.g. favouring access by product category, rather than geographic regions), the relational model advocates a methodology of data modelling and "normalization" to define multiple two-dimensional tables. When the data from multiple tables are required, the tables are "joined" to create a new two-dimensional view.

Figure: Star-Join Query

Queries to relational databases are constructed with Structured Query Language (SQL), an English-like computer standard. Many software products now offer a graphical depiction or prompted interface to simplify specification of the query, some with a data dictionary to translate requests into SQL. In contrast to alternative database models based on keys, the "non-navigational" orientation of SQL means that the user specifies the characteristics of the data to be selected, rather than the physical path by which the search should be conducted. Further, a "logical view" (e.g. a weekly accumulation) may be created by the database administrator as an abstraction of the true physical records (e.g. actual day-by-day transactions), and be similarly queried using SQL. Advances in distributed relational databases now enable queries to a local database to be forwarded on to a remote database, when the data are not available on the local machine.

Relational databases are well-suited to the structural evolution required of MDSSs. Instead of having to reorganize a database for each new view required, multiple tables may be joined in manners originally unforeseen. Although some new technologies such as object-oriented databases[16] may better handle complex datatypes (e.g. images of printer advertising), it is likely that the bulk of marketing data available (e.g. point-of-sale unit and dollar movements and inventory) will remain in relational tables. Improvements in the management of the ever-increasing volume (i.e. terabytes) of data will come as the result of incremental insight on how the data are to be used, and simpler methods of depicting the structure of data to end users.

[16]Object-oriented databases are currently in use for CAD/CAM and multimedia applications. For a deeper comparison of the relational and object-oriented strcutures, see Christopher M. Stone and David Hentschel, "Database Wars Revisited", Byte, October 1990, pp. 233-242; Herb Edelstein, "Relational vs. Object-Oriented", DBMS, Volume 4, Number 2, November 1991, pp. 68-79; or Won Kim, "Object-Oriented Database Systems: Strengths and Weaknesses", Journal of Object-Oriented Programming, July/August 1991, Volume 4, Number 4, pp.21-29.

The second key technology, **workgroup networks**, currently appears in marketing organizations as computer hardware which has often not been fully exploited by software. A bare-bones implementation is characterized by a network adapter card installed in a personal computer, so that the user sees virtual disks (e.g. an H: drive, when there is physically only one C: drive on his immediate machine) from which programs and data may be copied. More sophisticated workgroup software conceals the physical location of the data and programs, and represents other computers as extensions of the marketer's own. In an environment of structural evolution, it is important not only to be able to access the same data as co-workers, but to also tap into the models and methods which comprise the expanding knowledge base of the organization.

The simplest software environments to implement are **peer-to-peer** architectures, where all computers on the network are similar in hardware and operating system configuration. Common functions include shared printers, access to site-licensed software packages, and simple mail facilities. These are an electronic alternative to the "sneaker-net", where data and programs are replicated on diskettes, and then physically carried to another machine for copying. More advanced workgroup functions include electronic conferencing, as repositories to exchange knowledge; and workflow processing, to ensure the execution of interdependent tasks. In a team of marketers, an MDSS should not only assist the individual, but also enhance communications within a group.

Client-server architectures enable more sophisticated "program-to-program" communications between a client machine (e.g. a personal computer requester) and a centrally-maintained server, (often a system of a different platform, e.g. a minicomputer or mainframe). Advanced client-server programs can transparently extend the personal computer, such that external databases, communications to remote systems, and/or mail from other locations seamlessly appear to be part of the user's personal system. "Cooperative processing" between the two computers balances tasks to each's strengths, coordinated through short synchronizing messages. Commonly, the graphical user interface is the Q/A front-end in the personal computer client, and computationally-intensive routines or access to large volumes of data are managed by the server. In client-server decision-support systems, marketers are often responsible for assembly of their own models and optimization routines on the client systems, while the I/S departments manage the data and statistics in the servers.

The third key technology, **User Interfaces** (and Graphical User Interfaces -- GUIs[17] -- in particular) has changed the face of computing. For

[17]The Apple Macintosh environment, and Microsoft Windows (on top of PC-

users, mass-marketed, shrink-wrapped packages such as spreadsheets provide a capability to personally perform simple calculations and formatting. More involved procedures, as might be experienced in developing marketing models, however, often test the bounds of capabilities for which the package was designed. The traditional path for growth, custom application development, is undesireable for MDSSs. A second alternative, combining several specialized packages together, is difficult in most GUIs. A "cut-and-paste" or "clipboard" approach visually copies data from one software package to another, but often loses the richer attributes of the data (e.g. numbers that add up), causing user to thinking about datatypes (e.g. numeric, text or graphic) rather than the meaning of the data itself.

The popularity of GUIs is driving some uniformity in the "look" of software, but the "application-orientation" of packages from different vendors can give each a slightly different "feel". As feature upon feature is added to each package, users may solve problems by using the software they know best, rather than choosing the most appropriate method. In a semi-structured environment such as an MDSS, the risk may be best expressed by the maxim: "When the only tool you have is a hammer, every problem looks like a nail". The recent trend in software development has been to create environments where packages focus on their primary strengths, and packages may be more simply "snapped together" as the user desires.

Object-Oriented User Interfaces (OOUIs) provide a more consistent foundation for packages, by establishing a user's conceptual model of metaphors (e.g. a file drawer) and implementing an object model with appropriate behaviours (e.g. dragging another object onto a file drawer)[18]. Some software environments currently offer "macro" facilities which can capture keystokes and replay them by rote[19], but these approaches have not proven robust for "industrial-strength" applications. The next generation of operating systems, currently in development, will provide object-oriented characteristics which will enable non-programmers to "encapsulate" their procedures and visually "link" objects on their screens[20].

DOS) are popular examples of GUIs. Third party software developers build their application software proeucts to work on top of these environments.

[18]Abridged from David E. Liddle, "What Makes a Desktop Different", Metaphor Computer Systems, September 25, 1989.

[19]The "Agent" in the Hewlett Packard New Wave environment creates "scripts" of keystrokes and mouse clicks for later invocation. New Wave is designed as an application environment for Microsoft Windows GUI, which is in turn based on the MS-DOS operating system. See John R. Rymer, "Unraveling the New Wave Confusion", Patricia Seybold's Office Computing Report", Vol. 14, No. 9, September 1991, pp. 3-14.

[20]"Application Capsules" in the Metaphor Data Interpretation System have

Specialized routines (as demonstrated by the functionality of CoverStory) may eventually be purchased as "**components**"[21] which may be simply connected together to other components which have already been assembled. In this environment, marketers will have a greater facility to control the structural evolution of their models, without having to write programming lines of code.

The three technologies discussed above have each evolved independently, but it is the combination of the three which provides an environment suitable for an MDSS. Data repositories based on multiple relational tables enable marketers the flexibility to rejoin normalized data structures in the manner which is required at the moment. Workgroup networks enable the data to be physically located on any machine within or outside the marketing team, and facilitate the communication of marketing knowledge and models in addition to the data. Object-oriented user interfaces will provide the marketer with the "glue" to assemble components of data and programs into marketing models. This environment will provide a foundation for adaptive design, conceptual evolution and structural evolution, as described by Keen. Although it is possible to use these technologies to create a "black box", as defined by Little, they present the opportunities to provide a more open environment, where the marketing manager has a better access and understanding of the marketing scientist's thinking.

Clearly, adoption of these advances in information technology does not occur without strong leadership and vision. The next section discusses the management role in the evolution of marketing decision support systems.

6. Management's Role in the Development of MDSSs

The concept of Marketing Decision Support Systems is now over a

been used to direct data from one object to another (e.g. Query from a relational database, to a Spreadsheet, to a Text Tool) since the mid-1980s. More recently, Apple has implemented a slightly different concept with the "Publish/Subscribe" feature in their System 7.0. See Patricia B. Seybold, "Metaphor Computer Systems: A Quiet Revolution", Patricia Seybold's Office Computing Report, Vol. 11, No. 8, August 1988; and Tom Thompson and Owen Linderholm, "Seven's a Success", Byte, June 1991, pp. 42-48.

[21]Charles Irby graciously provided me with an unpublished "Constellation Technical Overview White Paper" in which Patriot Partners (founded by the IBM Corporation and Metaphor Computer Systems in 1990) described the development of an application environment where "Component" software would be interconnected by "Protocols". This mission was superseded by the formation of the Taligent partnership between IBM and Apple Computer, Inc.

decade old. What challenges face the marketer of the 1990s? Four areas which require some focus can be suggested: (1) **positioning** expectations for MDSSs, in comparison to other technologies; (2) development of "**model bases**" to capture "organizational knowledge", (3) improved **distribution** of the knowledge in the MDSS, and (4) the **ongoing renewal** of MDSSs.

Although many computer-based technologies have been loosely called MDSSs, it is important to **position** MDSSs neither too low nor too high in their capabilities. An MDSS may be used as report generator, just as a bicycle may be used as a delivery vehicle -- it may be simple to use for small tasks, but it may not be well-suited for industrial-strength jobs. MDSSs are unlikely to replace large, production computer systems optimized for efficiency on routinized transactions. At the other extreme, there is some controversy over how much "intelligence" can be built into an MDSS. To the novice computer user, there may appear to be little difference between one "black box" called a "decision support system", and another called an "expert system". Once the input data and algorithms are in place, is it a small task to replace human judgement? By Keen's definition, this change would require the complete structuring of a "semi-structured" problem, and closing opportunities for adaptive design, conceptual evolution and structural evolution. Such a system, incorporating human judgement, would then be called a knowledge-based system (KBS), rather than a decision support system. Unfortunately, the software development process for knowledge engineering leave little room for the marketer to incorporate his/her learnings, once the system is in production. Today's KBS shells are designed to create complete, production systems, and do not interface well with other operating environments. In the emerging object-oriented world, (perhaps in the beginning of the next millenium), it will be challenge to see whether the "knowledge bases" and "inference engines" can be "componentized" to work with other software functions.

In the more immediate future, how then can MDSSs "grow" in their contribution to the business? Much as a "marketing data base" has become a standard within many companies, a "**marketing model base**" must be developed. The most analytical marketers reviewing marketing data are skilled in identifying important events in the marketing data, and focus their analyses to support actionable decisions. Models are developed in their heads, and applied with experience. In order to improve the skills of the less-sophisticated marketers, some of these procedures must be captured in the MDSS software. Who should take responsibility for populating the model base? Unfortunately, marketers with day-to-day responsibilities do not usually capture the process of "adaptive design" in the software.

Since the supply of ready-to-hire marketing scientists is likely to remain below the level of demand, for the near future, cross-training the current workforce may be the only reasonable option to obtain the desired

combination of skills. Marketers with computing skills and programmers with a business sense are obvious choices, although both usually require development of their use of statistics. The type of system used by these "experts" is also an area of controversy: given the greater analytical requirement and understanding of the marketing scientists, should they have an "advanced" system, while "novices" are given a "simple" system? With the increased popularity of object-oriented user interfaces over the next few years, it will become more common to "encapsulate" complex analyses, so that an "average" marketer will not have to see what is "under the covers", unless he wishes to do so. In the new generation of "componentized" software, the marketing scientist will build components which the average marketer can "snap in" to an analysis, giving some flexibility in "shaping the DSS". Companies which choose not to migrate from the previous generation of software, however, are likely to retain separate levels of sophistication for the expert and for the novice.

The **distribution** of marketing models as software components can simply be performed electronically, as local area networks (LANs) have become more common. "Public File Drawers" are shared resources accessible to the workgroup, as a natural extension of their own desktop. It is not the physical distribution of marketing models which presents the challenge, however, but instead, the view of organizational knowledge as a corporate resource. Keen makes a distinction between:

> **Personal Support Systems (PSS)**, for use by individuals in tasks which involve no interdependencies, so that the user can indeed make a decision;
>
> **Group Support Systems (GSS)**, for tasks with "pooled" interdependencies which thus require substantial face-to-face discussion and communication; [and]
>
> **Organizational Support Systems (OSS)**, for tasks involving "sequential" interdependencies.
>
> A PSS may thus support a manager's own budget decision, a GSS support the budget negotiation, and an OSS support the organizational budget process[22].

Marketers dissatisfied with their computer support may have originally turned to personal computers as PSSs, but the need for individual marketers to share a centrally-managed database signals the infrastructure of an OSS. The era of personal computing is coming to a close, and the trend towards

[22]Keen (1980), pp. 5-6.

downsizing corporate staffs and flattened marketing organizations[23] calls for improved teamwork through the sharing of methods and knowledge.

By the very definition of MDSSs, **ongoing renewal** of the "data bases" and "model bases" is required to support decisions in the changing marketplace. If a problem becomes sufficiently structured, it should be considered for recoding in a traditional computer language with greater execution efficiency. New data sources are likely to become available, and methods of analysing existing data are sure to improve. The individuals with responsibility to support and maintain the marketing knowledge through all of these changes is unclear in many organization. The natural turnover of employees not only results in a loss of organizational knowledge, but presents a learning challenge as new marketers must follow in the footsteps of those preceding them. In order to consciously capture and retain organizational knowledge, a role for a "marketing methods librarian" may arise.

In closing, given the change in computer technology over the previous decade, should we prepare for another radical shift in the MDSS environment for the next decade? Although many might point to the shift of platform from the personal computer to the scientific workstation as the next trend, this is unlikely to have a great impact on MDSS users. The advent of the Graphical User Interface on a client/server architecture means that, to the average marketer, the MDSS will "look" the same. The ability to link "componentized" software will modularize the applications created by marketing scientists. Programmers may have to adjust to object-oriented technology, but this will probably be hidden from the marketers. In the next phase of evolution of Marketing Decision Support Systems, the emphasis will not be on technology, but on the building models which will assist marketers in making better decisions.

[23]See George Low, "Conference Summary: Conference on Sales Promotions from the Consumer, Manufacturer and Retailer Perspectives", Marketing Science Institute Report #92-103, February 1992.