NISTIR 88-4002

Data Management Strategies for Computer Integrated Manufacturing Systems

Wayne Davis

Department of General Engineering
University of Illinois
Urbana, Illinois 61801

Albert Jones

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Center for Manufacturing Engineering
Gaithersburg, MD 20899

Sudha Ram

Department of Management Information Systems
University of Arizona
Tucson, Arizona 86721

November 1988

Issued June 1989

NISTIR 88-4002

Data Management Strategies for Computer Integrated Manufacturing Systems

Wayne Davis

Department of General Engineering
University of Illinois
Urbana, Illinois 61801

Albert Jones

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Center for Manufacturing Engineering
Gaithersburg, MD 20899

Sudha Ram

Department of Management Information Systems
University of Arizona
Tucson, Arizona 86721

November 1988

Issued June 1989

National Bureau of Standards became the
National Institute of Standards and Technology
on August 23, 1988, when the Omnibus Trade and
Competitiveness Act was signed. NIST retains
all NBS functions. Its new programs will encourage
improved use of technology by U.S. industry.

U.S. DEPARTMENT OF COMMERCE
Robert Mosbacher, Secretary

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Acting Director

DATA MANAGEMENT STRATEGIES FOR COMPUTER INTEGRATED MANUFACTURING SYSTEMS

Wayne Davis
Department of General Engineering
University of Illinois
Urbana, Illinois 61801

Albert Jones
Center for Manufacturing Engineering
National Bureau of Standards
Gaithersburg, Maryland 20899

Sudha Ram
Department of Management Information Systems
University of Arizona
Tuscon, Arizona 86721

ABSTRACT

A recent worldwide trend to improve productivity in manufacturing has centered around the adoption of computer technology. Efforts are underway in many plants to use that technology to automate and integrate all manufacturing functions. This is transforming those plants into computer integrated manufacturing (CIM) systems. This paper addresses some of the special problems that have been and will be encountered in designing data management strategies for CIM. It describes both the major manufacturing functions themselves and the data required to carry out those functions. It also includes discussions on the various alternatives for data placement, data modeling, data administration, and data communication for CIM.

1. INTRODUCTION

Most major manufacturing companies have made a strategic decision to make extensive use of computer technology in their factories. In the beginning, computers simply collected data and provided information to human decision makers. Now, the computer itself makes many of those same decisions. The short term effect has been to improve the productivity of many individuals and the quality of their work. The long term goal is to have computers play a pivotal role in automating and integrating every phase of manufacturing. This computer integrated manufacturing (CIM) system is expected to produce higher quality at reduced costs, thereby improving the competitiveness of the entire organization.

The process of automating all major manufacturing functions and integrating them into a successful CIM system is proving to be difficult and time consuming. To date, researchers have concentrated primarily on the automation aspects of this process. Several computer hierarchies [Dav84, Jac87], similar to the organizational hierarchies that exist today, have been proposed. Various functions are assigned to each level within the hierarchy, and interfaces between the levels are specified. We believe that developing the tools required to provide the data that these computers need to perform their assigned functions will be the key ingredient in the integration aspects of that process.

In this paper, we discuss several obstacles to effective data management in a CIM environment and some potential techniques for removing those obstacles. The paper is divided into four major sections. In the next section we describe six major manufacturing functions which are depicted with their associated information flows in Figure 1. In sections three and four we examine the characteristics of the data needed to carry out those functions and the impact of the CIM environment on data management decisions. In section five we discuss 1) some approaches to resolving the issues introduced in sections three and four, and 2) some of the important unresolved optimization problems in the area of data management for CIM. Finally, we include a summary and bibliography.

2. MAJOR MANUFACTURING FUNCTIONS

As noted above, CIM requires the integration and, to the extent possible, automation of all major manufacturing functions including (1) marketing and sales, (2) manufacturing data preparation, (3) production planning and inventory control, (4) production scheduling, (5) process supervision, and (6) quality assurance. To understand the difficulty involved in achieving this goal, it is necessary to understand just what these functions do.

2.1 Marketing and Sales

Marketing and sales provide the primary interfaces between a manufacturing facility and its customers. They inform customers of available products, generate orders for selected products, price those products, negotiate delivery schedules, track shop floor performance in meeting those schedules, and ensure customer satisfaction after delivery. They also assist the customer in producing specifications for new or improved products. And, they often conduct a needs analysis to determine potentially profitable new products.

2.2 Manufacturing Data Preparation

Manufacturing data preparation includes all of the functions required to generate the data needed to manufacture a product capable of meeting a particular customer's requirements. These functions are usually assigned to engineering and process planning. Engineering translates a set of customer requirements into product designs which include detailed 3-D drawings, geometry data, tolerances, and other required manufacturing specifications. The emphasis on "design for manufacturability" is changing the way in which this function is being performed [Cha85]. These designs are then used by Process Planning to generate a complete list (including any possible alternatives) of raw materials, tools, machines, fixtures, and the precise machining instructions to be used during the entire fabrication process.

2.3 Production Planning and Inventory Control

Production planning is responsible for developing a list of "jobs" to be done on the shop floor during the next planning horizon (usually several months). In addition, it determines the hardware and materials necessary to do those jobs. This is accomplished in two steps. First, aggregate production planning (APP) uses both the current and projected demands established by marketing to set production quotas and inventory requirements for each product type during each of several smaller time periods (usually one week) during the chosen planning horizon. Those inventory requirements include all raw materials, tools, fixtures, castings, forgings, etc. needed to meet the demands. The APP continually monitors and updates production quotas and inventory policies based on the feedback from the DPP and updated demand forecasts from marketing.

Detailed production planning (DPP) uses these assigned quotas to generate production and inventory "jobs" for each time period. Before a production job is released to the shop floor for scheduling and processing, it is assigned a priority and a due date, and a check is made to verify that the required materials are on hand. Looking at future production quotas, the DPP may issue jobs to external vendors to replenish inventories. The DPP monitors the differences between the assigned and anticipated job completion dates, and if needed, changes both the due date and the specification of the criteria to be considered in the scheduling function.

2.4 Production Scheduling

Production scheduling develops detailed (usually daily) schedules of the operations required to complete the jobs issued by the DPP. These operations are then assigned to the various processes together with their anticipated start and finish times. Due date performance may be but one of several criteria to be considered in establishing the sequence of activities. The DPP provides the primary input on which criteria are to be considered and the desired compromise strategy to be employed in making tradeoffs among the criteria.

Once a production schedule has been generated, it is necessary to coordinate activities at each process to ensure that the schedule is met. This inter-process coordination function (IPC) requires continuous monitoring of the feedback from process supervisors. This feedback allows the IPC to ensure that all required materials have arrived at the process before the stated start times and to determine if the process will finish its assigned tasks at the anticipated finish times. This information is then used to update the existing schedule.

2.5 Process Supervision

Each process has a supervisor who has two responsibilities. First, he will implement the precise instructions from the process plan for every assigned operation. Second, the supervisor must then monitor the process during its execution of that operation to verify conformance to those instructions. Monitoring is typically sensor-based and allows the supervisor to detect changes in the processing environment. He can compensate for minor changes without substantial deviations from the original instructions. But, major problems will often force him to wait for a new set of instructions from process planning and the IPC before completing the assigned task.

2.6 Quality Assurance

Quality Assurance (QA) is divided into two major functions. First, it verifies that the output from each process meets the specifications prepared by engineering. These checks are the result of both on-line and off-line inspections. Whenever errors are detected, this information is used to correct problems in the designs, the process plans, and the processes themselves. Second, QA keeps historical records which can be used to improve the quality of all phases of the manufacturing system. In some cases, these records take the form of statistical studies which are used to track past and predict future equipment performance. These studies help guide decisions regarding machine maintenance and tool replacement. In other cases, information is archived on each product. This includes CAD designs, process plans, inspection and machining procedures, and other materials used in the fabrication of the that particular product. This helps guide decisions regarding that product the next time it is manufactured.

2.7 Remarks

The main purpose of a data management system is to provide the users of that system - computer processes and human beings - with access to the data they need to carry out their assigned functions. The design and implementation of such a system is always affected by two factors: the data and the characteristics of the interactions among that data, and the environment that must be supported by that data. These topics are addressed in sections three and four respectively.

3. THE DATA

The major pieces of data needed to support the manufacturing functions discussed in section 2 are now described.

3.1 Marketing and Sales Data

Marketing and sales provide the primary interface with the customer. To perform their assigned functions, they must retrieve and update information in numerous databases. These include product catalogs, customer orders, both the current and projected manufacturing capacities, finished products inventory, schedules, anticipated completion and delivery times, and orders for raw materials.

3.2 Manufacturing Data Preparation Data

The manufacturing data preparation function translates the customer's product specifications into a process plan. First, specifications are used to construct computer aided design (CAD) drawings. In some cases, data files containing those drawings are provided directly by the customer. In other cases, the drawings are generated from other customer supplied information. But, since CAD drawings do not always capture the entire set of specifications, it is often necessary to supplement those drawings with additional design detail. This information is attached to the CAD data file and stored in the database for later retrieval by process planning.

Process planning uses the that file to 1) develop a sequence of operations to meet the design requirements, and 2) describe the processes that will be needed to perform each operation. Today, this is largely a manual task with some computer assistance. It requires a great deal of human expertise and significant interaction with the database. A three step procedure is used. First, the product is given a group technology classification code [Cha85]. This code is then used to retrieve existing plans for products with similar processing requirements. Finally, a process plan for the new product is created by revising, and possibly merging, one or more of these other plans. A test part is then made from this plan to verify that it meets all the specifications. If it does not, then the plan is changed and retested. This scenario continues until a correct part is made and the corresponding version of the process plan is placed into the production database. Although there is a trend to automate many of the tasks currently performed by the process planner, this basic three step approach will remain for the foreseeable future.

3.3 Production Planning and Inventory Control Data

As noted above, this function has two major components: aggregate (APP) and detailed (DPP) production planning. APP must access the following databases: actual and forecasted demand; processing requirements for each of the products that make up that demand; current inventory status on tools, finished goods, work-in-process, and raw materials; and projected shop floor capacities. APP uses that data to update two additional databases. The first contains the number of each product type to be produced in the next planning period. The second includes the orders for new tools, raw materials, and any other items needed to produce those products. This latter database is also updated whenever orders are filled, cancelled, or changed.

DPP must access the databases updated by APP together with those containing (1) process durations and precedence relations for each product to be produced and (2) detailed information on process utilization. The former is typically part of the process plan. The latter includes uptime, planned downtime, and any other restrictions on availability. The DPP uses this data to update release dates, priorities, and due dates for each job issued to the shop floor and requested availability times for any required but still outstanding inventory.

3.4 Production Scheduling Data

The production scheduler (PS) is responsible for maintaining an accurate schedule of activities at all processes on the shop floor. That schedule must be updated whenever (1) a new job list is received, (2) an existing job is cancelled, finished, or given a priority update, and (3) a process experiences an unexpected delay. When a new job is assigned, the PS must first determine all of the required operations and the processes (and any alternatives) preferred to perform those operations. To do this, the PS must retrieve both the job list created by the DPP, and the process plan for each entry on that list. Then, the PS updates the current schedule by inserting the start and finish times for these new operations. To do this, the PS retrieves the current schedule and process utilization databases and executes some scheduling algorithm. Similar tasks are required to handle priority updates and unexpected delays. Simple database updates can be used to deal with cancelled or completed jobs.

3.5 Process Supervision Data

Each process supervisor monitors the execution of the operations assigned to the process under his control. To do this, he must access to several databases: the list of assigned jobs and scheduled start and finish times, their associated process plans, Numerical Control (NC) code or other equipment level programs, part description data, tool data, and fixture data. After each job has been completed, the supervisor must update the job database indicating the exact operations performed, the total processing time in the scheduling database, and the equipment and tool usage in their respective databases.

3.6 Quality Assurance Data

Quality Assurance (QA) tracks both the short term and long term quality of all manufacturing operations and the products they produce. Short term QA is achieved through on-line and off-line inspections of both equipment and products. This requires access to inspection plans, equipment usage charts, and planned maintenance schedules. Once the product inspection has been completed, the product history and scheduling databases must also be updated. Usage charts must be updated to indicate the total time every piece of equipment was used in the fabrication and inspection of each product. Long term QA is achieved through updates to all historical and maintenance databases.

4. THE ENVIRONMENT

The following sections examines several characteristics of CIM which impact the design and implementation of the system which must manage all of the data mentioned above. Similar discussions can be found in [Bar86, Su86].

4.1 Integration of Heterogeneous Systems

A CIM database is likely to be physically distributed across a network of heterogeneous computer systems. Individual systems will have a wide range of data access and data sharing capabilities. Some may only have file transfer mechanisms, while others may have sophisticated database management software. The software system which manages the full integrated database must mask the differences inherent in accessing data under these conditions. A common data manipulation language (DML) is needed to provide a standard format for all users to make database queries. Command translators are required to translate the queries posed in the common DML into the commands which can be executed by the data management software on component systems. Translators are also required to translate the type, format, and structure of the data interchanged among those component systems.

Finally, the management system must then be able to perform final assembly and formatting of the data retrieved from different component systems and deliver the result to the user in the desired form.

4.2 Evolving Nature of the System

The computer and production equipment which makeup these CIM systems will also be purchased from a variety of vendors over a long period of time. This modular expansion is expected for several reasons: to reduce costs, to increase flexibility and capability, to take advantage of evolving technologies, and to get away from the limitations of existing turnkey systems. This will have a significant impact on the design of the data management system. First, it should respond to user requests a manner which is completely transparent to that user. That is, the user simply requests and receives data; he is completely unaware of the effort required to satisfy that request. Second, the paths used to deliver data should be constructed by the management system at the time a requirement for data is identified. Third, it should allow frequent and dynamic updates to the data directory which contains, in part, the information about current system configuration and data delivery paths. Finally, it should include the software necessary to handle the network reconfiguration, replication and integrity control, and integration testing required to support this ongoing evolution.

4.3 Real-Time Operations

The systems that control shopfloor equipment have time-critical requirements for access to certain data. A variety of sensory and feedback data are used to make real-time decisions. However, some of this data may be shared by several users with different "real-time" access requirements. This implies that data may have to be replicated on several different systems and that updates must be made frequently and quickly. These updates require many conversions between 1) the data representation at the origin site, a system-wide data transport representation, and the desired data representation at the receiving sites, and 2) the local commands which execute a request and the global specification of the operations to be performed. In addition, the data management system must coordinate the global and local data directories and provide data and concurrency checks in a timely manner. Finally, this implies the existence of several sophisticated scheduling algorithms for both the data and network managers.

4.4 CIM Data Characteristics

The CIM data described in section 3 have significantly different characteristics from those found in the traditional business application database models. We give some examples below; more details can be found in [Su86].

system in various ways. It is a major factor in the decision to distribute or centralize data. It influences the choice of topology, protocol, and packetizing strategy for the network. Finally, it plays an prominent role in scheduling responses to data requests and for performing query optimization.

5. APPROACHES TO DATA MANAGEMENT

To completely specify a database management system (DBMS) which meets the needs raised in the preceding sections, we must address four major issues: data placement, 2) data modeling, and 3) data administration.

5.1 Data Placement

In evaluating data placement alternatives, we consider the tradeoff between reliability and storage costs as well as the tradeoff between retrieval and update costs. Retrievals are queries (see below) that only read data items from a database. Updates read, change, and write data values into a database. Four data placement alternatives are defined below.

Centralized - All data is stored at one node.
Replicated - All data is stored at every node.
Partitioned - Data is divided into disjoint segments and distributed across several nodes.
Hybrid - Some data is replicated and the rest is partitioned and distributed.

The following table indicates the relative measures from highest (4) to lowest (1) of storage cost, reliability costs, retrieval costs, and updates costs for each of these strategies. The costs can be expressed in terms actual expense, time, or complexity.

	RELIABILITY	STORAGE COSTS	RETRIEVAL COSTS	UPDATE COSTS
CENTRALIZED	1	1	4	1
FULLY REPLICATED	4	4	1	4
PARTIONED	2	2	3	2
HYBRID	3	3	2	3

TABLE 1-- DATA DISTRIBUTION ALTERNATIVES VERSUS DESIGN CONSIDERATIONS

4.4.1 Complex Data Objects. Complex data objects are those which are not readily defined in terms of atomic data elements. That is, any decomposition of such an object into purely atomic elements 1) causes a loss of meaning, and 2) places a heavy burden on each application by forcing it to manage all of the relationships and constraints for that object. One important example is the definition of a part. A part can be viewed as a series of intricate, machined features having complex topology and geometry relationships. Each of these features can then be decomposed into several levels of simpler, but related features. This decomposition continues until we have features that can be produced by a single machining operation.

4.4.2 Complex Data Types Complex data types such as set, vector, string, matrix, and ordered-set are common in manufacturing and should be treated as basic data types. In addition, operators should be developed which allow the data management system to manipulate these types directly. One example which may contain all of these types is the geometry model for a part. That model may contain (1) ordered-sets of named entities such as points, curves, and surfaces, and (2) multi-dimensional vectors and matrices.

4.4.3 Complex Relationships Very complex relationships can exist among manufacturing data. One example which exhibits several of those relationships is the history of a part as it proceeds through the factory. That history typically contains (1) an ordered list of the machines visited, the time spent there, and the operations performed there, (2) material transport directions including absolute and/or relative positioning data, grip points for robots, fixturing data, etc., (3) the process plan used to fabricate the part including intermediate (ideal) topology and geometry definitions, (4) summary statistics on all processing, cleaning, and inspection operations, and (5) any additional customer related data.

4.4.4 Recursive Structures Recursive structures are often the most efficient way of representing certain complex machining and inspection patterns. Complex features are one such pattern. A simple example would be a pattern of holes to be drilled for a part. This pattern can be defined in terms of the individual holes and their positional relationship. Each hole is then defined by its diameter, tolerance, and other special attributes.

4.4.5 Heterogeneous Data Traffic. As indicated above, manufacturing data comes in "all shapes and sizes". There are product catalogs, containing megabytes of data, which may be updated only once or twice a year. There are part models and process plans, which may contain several kilobytes of data, which are accessed and updated many times each month. And, there is equipment status data, which may be only a few hundred bytes of data, but which must be updated several times a minute. This heterogeneity of CIM data, together with the time constraints placed on delivering that data, impacts the data management

The objective is to minimize the costs of storage, retrievals, and updates, and to maximize the system reliability. The information in the table indicates that it is not possible to achieve all of these objectives simultaneously. For a particular application, the optimal choice depends on three factors: 1) the relative importance placed on achieving each of the individual objectives, 2) the total amount of shared data and the time constraints placed on that data, and 3) the cost of the data modeling, administration, and communication systems needed to support the data placement decision.

In general, as the level of automation and computerization goes up, so does the requirement for distributed data placement. The remainder of this paper assumes distributed data placement is the only meaningful strategy for CIM. Once this decision has been made, three additional issues must be resolved. The first involves the actual fragmentation scheme (see below) to use in allocating data to databases. The second involves the actual assignment of databases to physical locations and selection of a DBMS to manage that data. The third is replication. The data to be replicated must be identified, and the number and location of those copies determined. To date, these concerns, which can be formulated as optimization problems, have received little attention.

5.1 Data Modeling

As noted above, future manufacturing facilities will have some type of sophisticated distributed database system (DDS). A general architecture for a DDS is shown in Figure 2 [Cer84]. Each local database management system (DBMS) has its own data model containing 1) a Local Conceptual Schema to represent the logical relationships among the data elements contained in that database, 2) a Data Definition Language (DDL) for defining those schemas, 3) a Data Manipulation Language (DML) for expressing queries and for writing database programs. Each local DBMS also provides the capability of defining external views that are tailored for each group of users. Data is exchanged among these local DBMSs using a global data model, a data directory (which identifies the location of each data item in the DDS) and a data dictionary (that captures and depicts the meaning of each data item).

The global data model contains a global conceptual schema that defines how the data are related and distributed in all the databases throughout the system. In addition, there is a global internal schema which provides the routing and mapping information for processing queries. As depicted in Figure 3, the global internal schema contains three components:

1) Fragmentation Schema: The Global Schema is typically split into several fragments. The fragmentation schema defines the mapping between the Global Conceptual Schema and its fragments.

Each fragment is a logical portion of the Global Schema which describes how a schema is horizontally and vertically partitioned.

2) Allocation Schema: Each fragment may have two types of data instances: original copies and replicas of data from other sites. The allocation schema tracks the location of all replicas so that updates can be performed correctly.

3) Mapping Schema: The mapping schema maps elements of each fragment in the Global Conceptual Schema to the correct representation and structure for each local conceptual schema.

Horizontal partitioning refers to the fact that multiple sites may have their own subset of a logical record set. Vertical partitioning implies that a logical record may be divided into multiple subrecords which exist at several different sites. The amount and type of partitioning has a great impact on the data administration functions.

5.3 Data Administration

A data administration system has two major responsibilities. First, it must maintain the integrity and autonomy of all local databases. Second, it must ensure delivery of requested data in a timely and completely transparent manner [Bel84]. That is, a user should simply query the global database, and receive an accurate and timely response. He should be totally unaware of the effort required to answer his query. The subsequent sections describe the functions required to achieve those goals. They are followed by a discussion of design alternatives for implementing such a data administration system (DAS) in a CIM environment.

5.3.1 Concurrency Control: Recall that some of data will be replicated at more than one site. Consequently, the DAS must maintain consistency among the different copies of those data items while allowing concurrent access to them. Four potential problems must be addressed:

1) two transactions are simultaneously updating the same data item,
2) one transaction is reading an item while another is updating the same item,
3) two transactions requiring the same data items are waiting for each other (deadlock),
4) one transaction is continually preempted by others requiring access to the same items (livelock).

Methods for resolving those problems are known as concurrency control mechanisms (CCM). Although they vary in implementation, most existing CCMs are based on two principles: Two Phase Locking (2PL), Timestamp Ordering (T/O) [Koh80, Ull82].

Techniques based on the 2PL approach use software flags called "locks" to ensure consistency among replicated databases. A three step process is used. First, a transaction obtains locks on all data items. Then, one or more operations are performed on those items. Finally, the locks are released. This requires a lock manager to coordinate locking and unlocking. These responsibilities can reside at a central site, or be distributed across several sites. While this automatically eliminates problems 1 and 2, additional mechanisms are required to avoid both deadlocks and livelocks.

The T/O technique tries to establish a serial order in executing transactions based on timestamps generated by the DAS. Each transaction is assigned a timestamp when it starts executing. Each data item is also assigned a read timestamp and a write timestamp. These three time stamps are used for ensuring consistency among copies of a data item. While this approach addresses all four of the problems mentioned above, there is considerable overhead involved in storing timestamps for all the data items in the database. It also increases the execution time for transactions requiring the same data items.

It should be noted, that, to date, neither of these techniques has been successful in meeting the demands of the CIM environment.

5.3.2 Query Processing: Each user can make requests, called queries, from the global database. Since the user is totally unaware of the location of the data needed to fill that request, it should be made in the global DML. If, however, the local DML is used, additional translations may be required. Since the data may be physically distributed across several sites, the DAS must process that query in several steps [Cer84]. First, the original query must be decomposed into subqueries which only require access to a single logical partition. The next step is to choose the physical copies which will provide data item values for each subquery. Next, each subquery must be translated into the language used by the DBMS at the chosen site. Finally, the subquery results must be aggregated and passed back to the user. This may require several data representation translations.

In designing this query processor, two important issues must be addressed. The first involves the series of translations into and out of the global and local DMLs. There are two choices: translate at the global level or at the local level. The former relies on the data administration system to do all translations. The latter requires standardization so that each underlying DBMS has a translator from its own local QL and DML into and out of the global versions. Since these standards do not exist at the present time, global translation is the only viable alternative.

The second issue involves a global strategy for executing both queries and subqueries. At any point in time, the data administrator will have a queue of queries to execute and a derived queue of subqueries to assign to each underlying DBMS. A global strategy contains procedures for 1) sequencing both of these queues, and 2) efficiently decomposing each query into subqueries. Although such procedures exist for individual DMBSs, few solution techniques are available for the distributed environment [Chu86]. And, even less is known about solving these problems in the CIM environment where tight time constraints and complex manipulations are the rule rather than the exception.

5.3.3 Recovery: Problems may occur in a DDS due to failures at individual sites, failure of communication lines between sites, transaction completed incorrectly, and partially completed transactions. Mechanisms must be built in the DDBMS to recover from these problems. Typically, these mechanisms must be closely tied to the CCMs described above [Goo77]. Regular backups are one way to provide a known consistent point from which to begin recovery. Automatic entry in a log or journal is another way to recover from crashes. Global checkpoints are used to reconstruct a database after a catastrophic failure. Global checkpoints refer to a set of local checkpoints performed at all sites in the DDS indicating an overall consistent state.

5.3.4 Security: Security problems may also arise in a DDS. When secure data is requested by a remote site, the data administrator must 1) determine whether or not that site has been authorized to receive that data, and 2) ensure that the transmission of that data is also secure. The former can be done by establishing an identification protocol and authorization codes for remote sites. In addition to identifying sites, identification mechanisms and authorization rules can be required for individual users as well as classes of users. The latter can be accomplished by providing encrypting and decrypting mechanisms for all secure data transmission between remote sites. To date, very little work has been done in this area.

5.3.5 DAS Design Alternatives: As indicated in the preceding sections, the DAS controls access to all data. There are two traditional approaches to DAS design [Sto77, Wil82]: centralized and distributed. Both allow distributed data placement, but manage this distribution through a single supervisory process or a peer-to-peer negotiation. Initial designs required identical underlying DBMSs on similar components. This assumption could be relaxed, but updates to data throughout the entire distributed system could not be supported [Lan82].

The major drawbacks to the centralized approach are single point of failure, unpredictable response times, and low system reliability and subsystem autonomy. These drawbacks can be eliminated using a distributed control approach. However, the negotiations inherent in totally decentralized control increases 1) the response time, making real-time operations difficult, and 2) the complexity of implementing integrity, concurrency control, and recovery.

Researchers at National Bureau of Standards and elsewhere [Bar86, Dee85, Iis83] have begun to address these issues. They are pursuing a hybrid DAS architecture for the heterogeneous CIM environment. Some functions are performed at every node within the system. They include manipulating local data, translating queries and data representations into and out of local form, and providing interprocess and network communications. Distributed management services are assigned to selected sites and a unique master site ultimately resolves global dictionary changes and update conflicts detected by the selected sites.

Some commercial software has recently become available which will support the distribution of data [Joh84, Neu84, ingres*]. While this is encouraging, there are still limitations which prohibit these products from being viable solutions to the problems inherent in the distributed CIM environment.

6. SUMMARY

Efforts are underway in many plants to use advanced computer technology to automate and integrate all manufacturing functions. This transformation to a CIM environment has revealed several problems in the design and real-time control of data management systems for CIM. In this paper, we have described both the major manufacturing functions themselves and the data required to carry out those functions. We have also discussed various alternatives for data placement, data modeling, data administration, and data communication for CIM.

ACKNOWLEDGEMENTS

The authors are indebted to Mary Mitchell, National Bureau of Standards, for her helpful comments in the preparing this paper.

7. REFERENCES

[Bar86] Barkmeyer, E., Mitchell, M., Mikkilineni, K., Su, S., and Lam, H., "An architecture for distributed data management in computer integrated manufacturing", National Bureau of Standards Internal Report, NBSIR 86-3312, January, 1986.

[Cer84] Ceri, S. and Pelagatti, G., "Distributed Database: Principles and Systems", McGraw-Hill Book Company, 1221 Avenue of the Americas, New York, NY 10020, 1984.

[Cha85] Chang, T., and Wysk, R., An introduction to automated process planning systems", Prentice-Hall, Englewood Cliffs, NJ, 1985.

[Chu86] Chu, W., (ed), "Distributed systems, Vol. II,: Distributed database systems", Artech House INc., Deham, MA., 1986.

[Dav84] Davis, W., "Decision-making and control hierarchies for Production Systems", Technical Report No. 145, PLAIC, Purdue University, West Lafayette, IN, 1984.

[Dee85] Deen, S., Amin, R., Ofori-Dwumfuo, G., and Taylor, M., "The architecture of a generalized distributed database system - PRECI*", The Computer Journal, Vol. 28, 1985, pp. 282-290.

[Goo77] Goodman, N. and Rothnie, J., "A Survey of Research and Development in Distributed Database Management", Proceedings of the Third International Conference on Very Large Databases, 1977, pp. 10-27.

[Iis83] "Integrated information support system (IISS)", materials Laboratory, Air Force Systems Command, Wright-Patterson AFB, Feb, 1983.

[Iso81] "Data processing - open systems interconnection - basic reference model", ISO Standard 7498, International Standards Organization, Geneva, 1981

[Jac87] Jackson, R. and Jones, A., "An Architecture for decision-making in the factory of the future", ORSA Interfaces Special Issue on Manufacturing, to be published.

[Joh84] Johnson, H., Baum, L., and Beaudet, R., "IPAD distributed database management facility-IDF: architecture specification", Boeing Computer Services Company, 1984.

[Kle76] Kleinrock, L., "Queuing systems volume II: Computer applications", John Wiley, NY, 1976.

[Koh80] Kohler, W., "Overview of Synchronization and Recovery Problems in Distributed Databases", Proceedings COMPCON80, Fall 1980, pp. 433-441.

[Lan82] Landers, T., and Rosenburg, R., "An overview of MULTIBASE", Second Symposium on Distributed Databases, Berlin, Sept, North-Holland, NY, 1982, pp. 311-366.

[Ma84] Ma, R., "A model to solve timing-critical problems in distributed computer systems", Computer, Jan., 1984, pp. 62-68.

[Map87] "General motors MAP specification 3.0", SME, Dearborn, MI, 1987.

[Neu84] Neubold, E., "Distributed database systems with emphasis toward POREL", IPAD II, Proceedings of a National Symposium, NASA Conference Publication 2301, 1984.

[Sto77] Stonebraker, M., and Neubold, E., "A distributed database version of INGRES", Proceedings of 1977 Berkeley Workshop on distributed Data Management and Computer Networks.

[Su86] Su, S., "Modeling integrated manufacturing data with SAM*", Computer, Jan., 1986, pp. 34-49.

[Ull82] Ullman, J., "Principles of Database Systems", Computer Science Press, Rockville, Md., 1982.

[Wil82] Williams, T., et al, "R*: An overview of the architecture", Proceedings of the International Conference on Database Systems, Jerusalem, June, 1982.

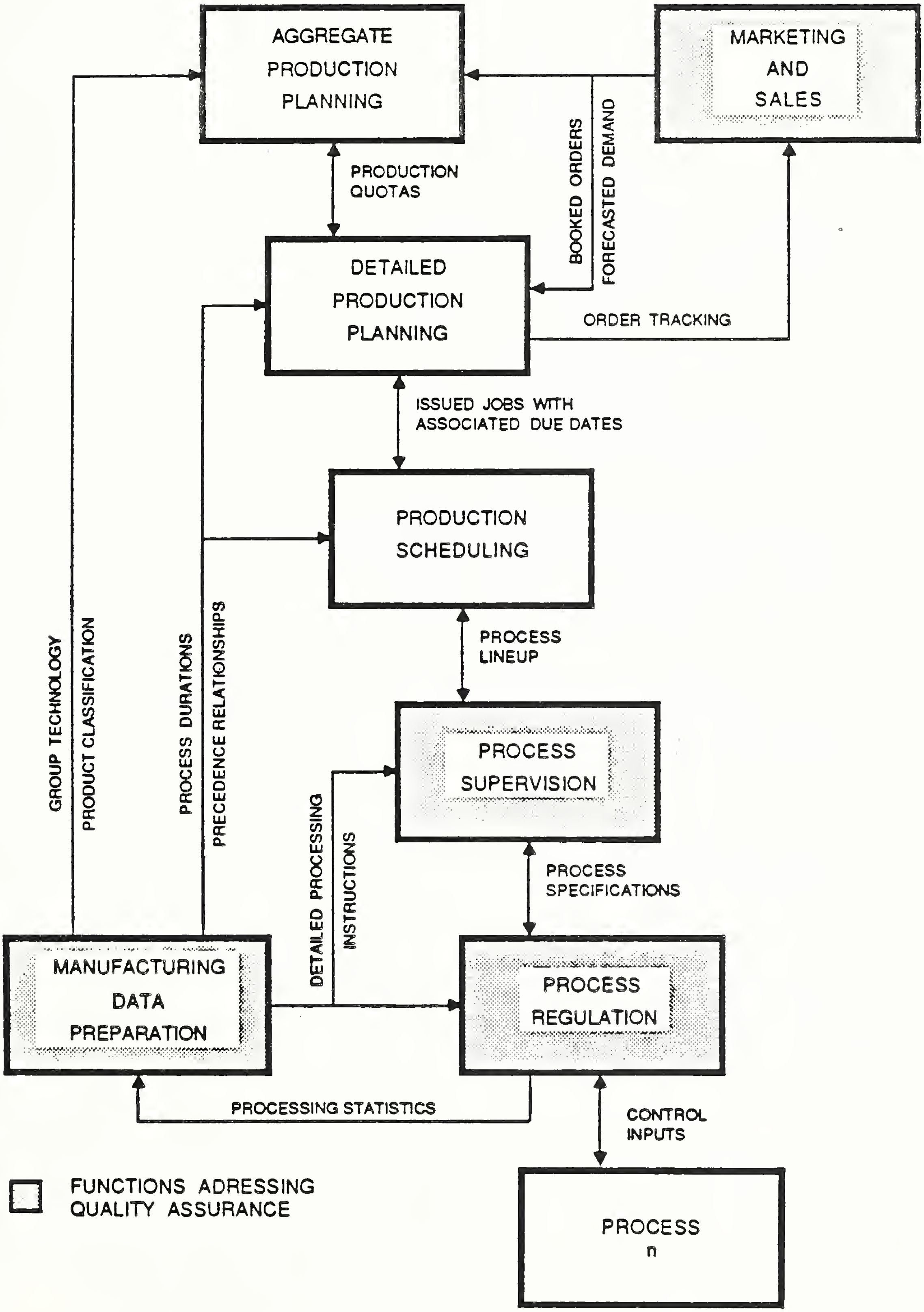

FIGURE 1-- MANUFACTURING FUNCTIONS COMPRISING CIM ENVIRONMENT

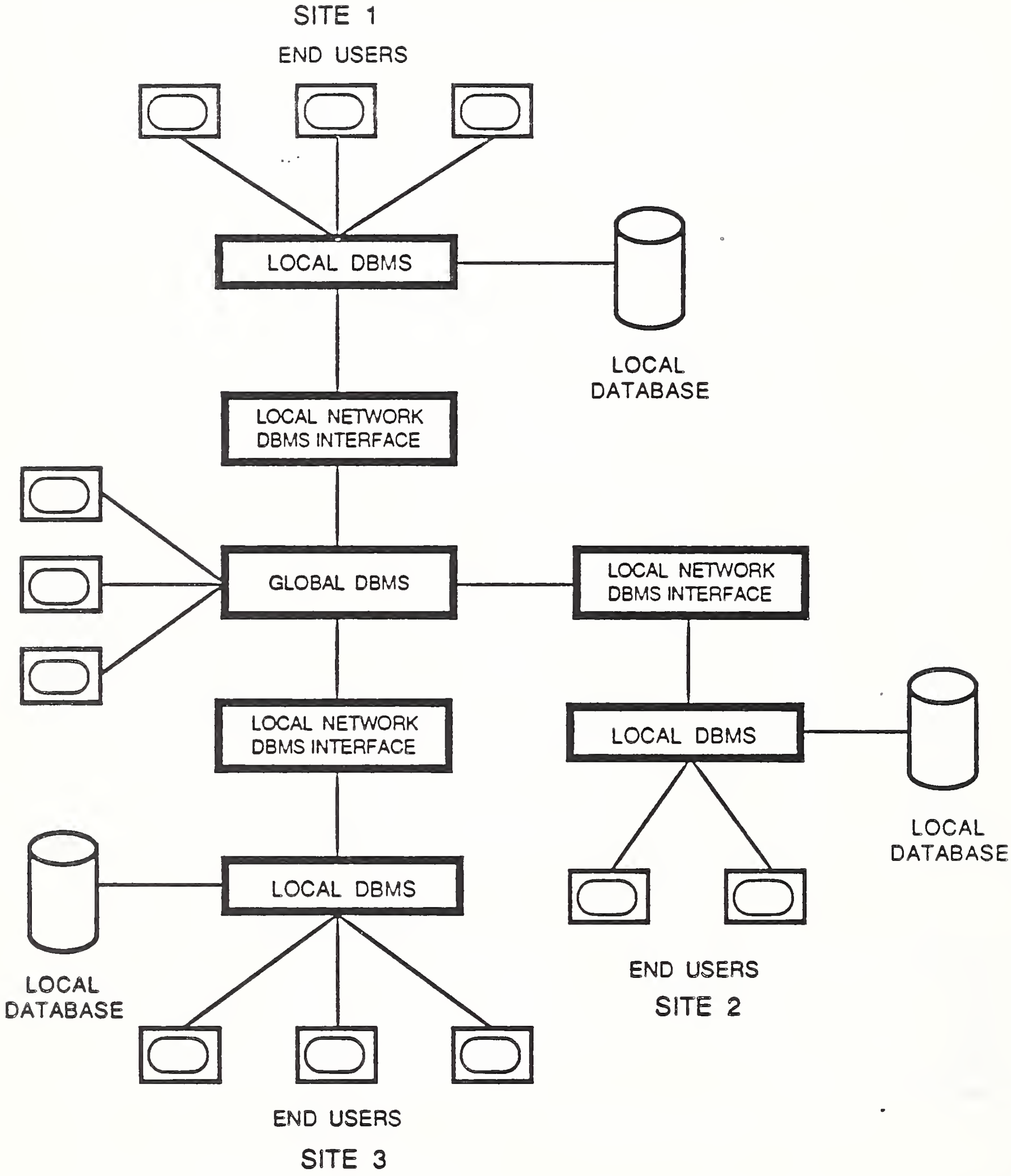

FIGURE 2--STRUCTURE OF A DISTRIBUTED DATABASE SYSTEM

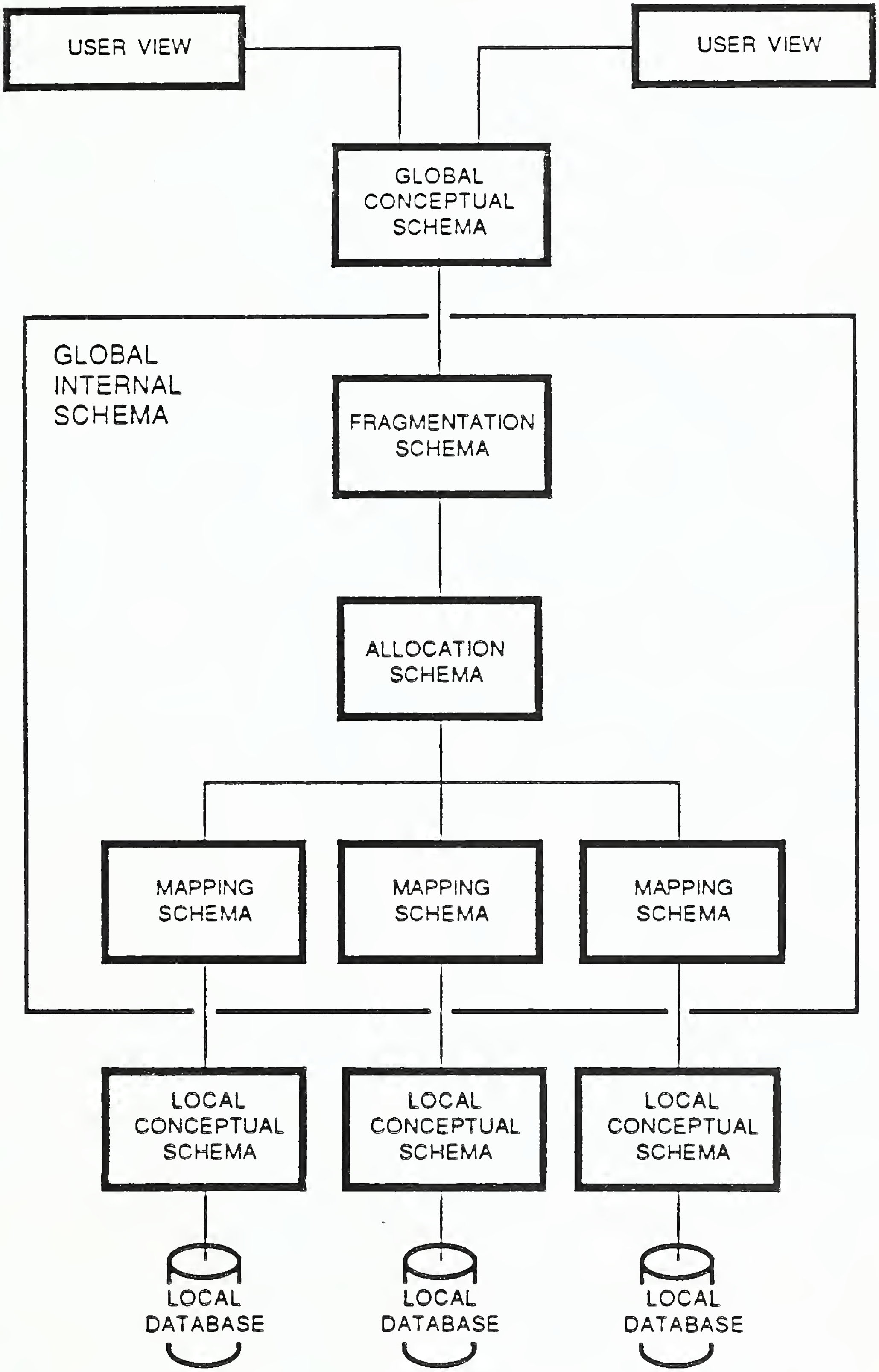

FIGURE 3 - - DISTRIBUTED DATABASE ARCHITECTURE

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NISTIR 88-4002	2. Performing Organ. Report No.	3. Publication Date JUNE 1989

4. TITLE AND SUBTITLE

Data Management Strategies for Computer Integrated Manufacturing Systems

5. AUTHOR(S)

Albert Jones, Wayne Davis, Sudah Ram

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

The National Bureau of Standards
Bldg. 220/B127
Gaithersburg, MD 20899

10. SUPPLEMENTARY NOTES

□ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

A recent worldwide trend to improve productivity in manufacturing has centered around the adoption of computer technology. Efforts are underway in many plants to use that technology to automate and integrate all manufacturing functions. This is transforming those plants into computer integrated manufacturing (CIM) systems. This paper addresses some of the special problems that have been and will be encountered in designing data management strategies for CIM. It describes both the major manufacturing functions themselves and the data required to carry out those functions. It also includes discussions on the various alternatives for data placement, data modeling, data administration, and data communications for CIM.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

Automation, computer integrated manufacturing, databases, data placement, data modeling, data administration, data communications.

13. AVAILABILITY

XX Unlimited

□ For Official Distribution. Do Not Release to NTIS

□ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

XX Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

23

15. Price

$9.95