Information Resource Management: Integrating the Pieces

By Salvatore T. March, Information & Decision Science Department, C. L. Carlson School of Management, University of Minnesota, 271 19th Avenue South, Minneapolis, MN 55455 and Young-Gul Kim, J. M. Katz Graduate School of Business, University of Pittsburgh, 222 Mervis Hall, Pittsburgh, PA 15260

Abstract

Information-based organizations depend upon computer databases and information systems for their ongoing operation and management. Information Resource Management (IRM) *is a program of activities directed at making effective use of information technology within an organization. These activities range from global corporate information planning to application system development, operation, and maintenance and support of end-user computing. Numerous approaches to specific IRM activities have been proposed. They remain disjoint, however, and, hence globally ineffective.*

A significant reason for inability to integrate IRM activities is the failure to adequately define the information resource. What is it *that must be effectively managed? This paper addresses this issue. It applies data modeling concepts to the problem of managing organizational information resources. A data model is developed to support and integrate the various IRM activities. This model formally defines the information resource and the data needed to manage it. It provides a basic ingredient for effective Information Resource Management.*

Keywords and phrases: Information Resource Management, data modeling, organizational information needs

A.C.M. Categories: K.6.4, K.6.3, H.2.1, J.11.

Research was partially funded by the McKnight Foundation.

INTRODUCTION

Information Resource Management (IRM) is a program of activities directed at making effective use of information technology within an organization [Fong and Goldfine, 1986, 1989; Diebold 1979]. These activities range from global corporate information planning to application system development, operation, and maintenance, and support of end-user computing. They include planning for and acquiring computer and communication technologies, selecting, implementing, and managing information system development methodologies, and re-engineering business systems as information systems are integrated into the business.

The need for IRM has been widely documented [Drucker, 1988; Edelman, 1981; Synnott 1987]. Information has become a valued and expensive corporate resource. Yet many organizations are unable to take advantage of this resource because it is often unplanned, ill-defined, and misunderstood [Gartner Group, 1990]. There are often redundancies and inconsistencies across organizational information systems, each having been designed for its own purpose and for its own set of users (IBM, 1981). It may be impossible to integrate information collected by these systems. The potential benefits of sharing organizational information resources are not realized.

The establishment of an IRM program implies significant cooperative efforts among functional business units. It must begin with strategic data planning and it must insure that the strategic data plan drives the development of information systems [Goodhue, Kirsch, Quillard, and Wybo, 1992]. Ownership of information must be replaced by stewardship of information. As plans are made and applications developed, business units who steward the information must understand and respond to the information needs of other business units. This implies significant changes in the way in which information systems are developed, funded, and evaluated.

Establishing an IRM program is itself a major business system development effort. It requires significant information system support. This paper develops a comprehensive definition of IRM using a data-oriented approach. As corporate data models provide the integrating mechanism for various information system applications [Scheer, 1989], the IRM data model provides the integrating mechanism for IRM activities. Methodologies focusing on IRM activities such as IS planning, application development, and data administration were analyzed in developing the IRM data model.

Data modeling has been successfully applied to developing information system applications [Carlis and March, 1984; Ross, 1987; Teorey, Yang, and Fry, 1986] and has recently been applied to analyzing organizational information needs [Eftimie and Nikles 1988; Olle, 1988]. In this paper we apply it to IRM.

The data model we develop defines the organizational information resource and the data needed to manage it. It is a "meta-data" model, that is, a data model of the data model used to describe the organizational information resource and thus to support IRM. It forms the basis for an Information Resource Dictionary System (IRDS) [Dolk and Kirsch, 1987; Law 1988; Mercurio, Meyers, Nisbet, and Radin, 1990; Sagawa, 1990]. It includes aspects of strategic information planning, application selection and development, and data management. It is the key ingredient for Computer Aided Planning (CAP) [Feuche, 1990]. It defines a central repository through which planning, development, and management of information resources can be coordinated.

This IRM data model is presented as a "straw man." Recognizing that the whole of IRM is extremely complex, we focus on IS planning and development. We hope that the presentation of this data model will raise issues and provide a forum for discussing the information requirements for effective IRM.

The data model does not dictate methodology. It describes what data must be gathered in order to accomplish IRM; it does not prescribe how that data is gathered. Multiple methodologies can be supported for different tasks. As various tasks are done (e.g., IS planning, system selection, feasibility studies, general and detailed system design, system construction), parts of the data model are populated. In defining what data must be collected in performing a task, the data model assures that this data is available to tasks that populate other parts of the data model. Furthermore it assures that the relationships among tasks and among systems are documented.

The remainder of this paper is organized as follows. Section 2 presents a brief overview of Information Resource Management. Section 3 develops a data model that defines and integrates the information requirements for IRM. In Section 4 we present conclusions and directions for future research.

INFORMATION RESOURCE MANAGEMENT

The focus of Information Resource Management (IRM) is the effective development, management, and utilization of organizational information. IRM encompasses all policies, procedures, and actions concerning organizational information systems [Fong and Goldfine, 1986; 1989]. It includes such diverse activities as strategic data planning (e.g., information, system, and technology architectures), capacity planning (e.g., long range technology planning), application selection, information system development and business system re-engineering, project management, hardware and software acquisition, and data administration.

Implementing an effective IRM program requires: (1) knowing the current state of the organizational information resource, (2) planning for its future development, and (3) controlling the activities related to its development and use.

Planning is based on an assessment of future corporate information needs relative to the current state of the information resource. Where there are disparities, activities to bring the current state into conformity with the needs are initiated (e.g., information systems are developed, technology is acquired, data is collected and managed). These activities must be controlled to insure that the needs are actually met and that these activities are accomplished in an effective and efficient manner. Thus, representations must be developed for: the current state, the planned objectives, and the ongoing activities to accomplish these objectives.

The current state of the information resource is an inventory of:

(1) **Implemented Databases and Applications:** what data exists (its semantics), where it is captured, and how it is maintained (including characteristics such as format, age, accuracy, integrity specifications, unit of measure, degree of summarization, etc. [Davis and Olson, 1985]);

(2) **Information Delivery Capability:** the ability to deliver information, including the (utilized and slack) capacities of: hardware, software, and network platforms, organizational structures and personnel to support (controlled) access to corporate data (e.g., Data Administration and Information Centers), and

(3) **Development Capability:** the ability to develop new applications and to integrate data from existing applications; this includes a measurement of the skill levels of development personnel (both MIS professionals and end-users) and the hardware and software tools available to support the development process.

Existing Data Dictionary Systems (DDS) partially address the inventory of implemented databases [Allen, Loomis, and Mannino, 1982; Navathe and Kerschberg, 1986]. However, they also tend to describe physical characteristics of data elements and files (data element name, length, data type, file size) rather than logical characteristics (entities, attributes, relationships) [Law 1988]. A more comprehensive inventory integrating both logical and physical data and processing capabilities is needed for IRM.

Planning for the development of the information resource requires a representation of:

(1) **Corporate Information** Requirements: often summarized in an Information Architecture matrix

[Wetherbe and Davis, 1983], that specifies the basic functions of the organization, the basic classes of data needed to accomplish those functions, and the interactions between data and function;

(2) **Delivery Disparity:** an assessment of the functions and data classes that are not sufficiently supported by the current information resource, including an assessment of the benefits that would be derived from such support, the importance of these benefits (measures can include expected dollar return as well as contribution to corporate objectives), and the cost of accomplishing this level of support (hardware, software, and personnel); and

(3) **Data Stewardship:** a corporate wide data policy defining responsibility for acquiring and maintaining this corporate data, including policies for establishing corporate wide data standards and allocating system development and operating costs.

Controlling these activities requires the establishment of milestones and feedback mechanisms so that progress against the plan can be measured. Three levels of control are needed:

(1) **Shared Data Capture:** applications that capture and manage data to be shared by other applications must conform to corporate definitions for this data including data integrity constraints, data retention, and in some cases even data values; control activities are aimed at insuring proper application design and implementation (including definition and sharing of common data capture modules);

(2) **Project Management:** measures progress in application development using system life cycle phases (adapted to accommodate prototype development approaches and user developed systems) and compares progress to milestones and expenditures;

(3) **Operation:** measures the conformity of acquisition of data to the definition within the data capture application(s) and monitors the use of shared data by application software and end-users.

To effectively accomplish these tasks we need a formalism in which to represent or model the information resources. Semantic data modeling provides such a formalism [Chen, 1976; Brodie, 1984; Hull and King, 1987; Peckham and Maryanski, 1988]. Using constructs such as: entities, attributes, and relationships, the data required for IRM can be represented.

The IRM data model must distinguish among: planned, actual, and "in-development" information resources. It must facilitate mapping among these stages of development. It must support: (1) high level data and function descriptions as used in planning activities, (2) computer oriented data and process descriptions for use by data administrators and application implementors, and (3) user oriented system descriptions for use by systems analysts and end users. The challenge is to model multiple levels of information representation and to provide mappings among them.

MODELING THE INFORMATION REQUIREMENTS FOR IRM

IRM is a complex process involving the planning, developing, controlling, operating, and utilizing of organizational information resources. It requires IS and business functional areas to understand and communicate about the information resource. The IRM data model provides standards for data gathering within the various IRM activities and establishes a vocabulary for describing the information resource. It can be used to specify when development may proceed from one activity to another (i.e., when the data required for the preceding activity is specified). It is the integrating mechanism for the various IRM activities.

In this section we present a data model for IRM. It is developed in four subsections and illustrated in Figures 1 through 5. Subsection 3.1 develops a data model for corporate information planning (Figure 1). It captures the essential features of a wide range of information system planning methodologies. Subsection 3.2 develops a model of currently implemented information resources (Figure 2). It includes both existing physical databases and the applications that manage and use them.

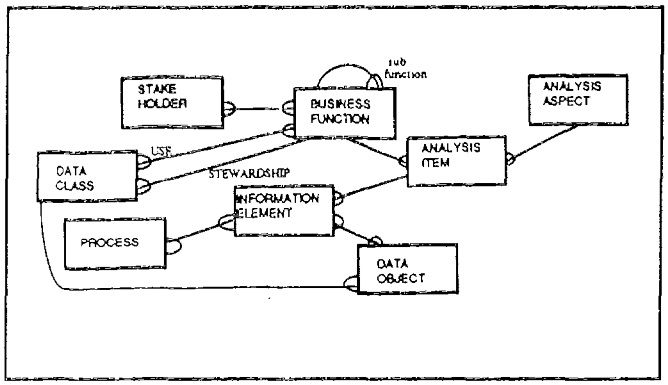
Noting the difficulty of mapping between the planning data model to the implementation data model, Subsection 3.3 develops a conceptual model of the content of the information resource (Figures 3 and 4). The model is conceptual in the sense that it is implementation independent. Thus it can act as a mapping vehicle that facilitates the integration of planned and actual information resources. Furthermore, it supports a wide range of information system development methodologies. Subsection 3.4 discusses this integration (Figure 5) and the resulting IRM control capabilities achieved.

A Data Model for Information System Planning

Numerous information system planning methodologies have been proposed (see, e.g., [Selig, 1982; Dickson and Wetherbe, 1985; Zachman, 1987; Shank, Boynton, and Zmud, 1985; Dooley, 1986; Lederer and Mendelow, 1987]). These are based on concepts such as: STAKEHOLDER, BUSINESS FUNCTION (and possibly sub-function), ANALYSIS ASPECT (such as problems, decisions, and critical success factors).

Figure 1 is a data model for corporate information planning. Each of the above concepts are entities ("things" about which information is maintained). They are represented by rectangles. Relationships between entities are represented by lines connecting the rectangles. To complete the representation we have also added the entities, ANALYSIS ITEM, PROCESS, DATA OBJECT and DATA CLASS (discussed below). Each entity also has attributes (characteristics describing the entity). To prevent the figure from becoming too cluttered, attributes are not shown.

Figure 1
A Data Model for Corporate Information Planning



Information system planning is typically done by a team of upper level managers representing the various business functions. Often it is facilitated by an outside consultant. The data model in Figure 1 is populated partially by the organization as it defines its IS planning methodology (see below); the remainder is populated during the information systems planning process. Capturing this data at that time and in this form assures that important facts discovered during the planning process are available when applications are selected and developed.

Referring to Figure 1, BUSINESS FUNCTIONs are the fundamental activities that must be performed in order for the organization to meet its objectives. A STAKEHOLDER is a member of the business organization who has a "stake" in the successful operation of one or more BUSINESS FUNCTIONs. BUSINESS FUNCTIONs and STAKEHOLDERs are typically identified by taking a general systems view of the organization and of organizational responsibilities (the facilitator asks, "what does the business do?" and "who is responsible/effected by it?").

Each BUSINESS FUNCTION has one or more STAKEHOLDERS. Hence, the relationship between STAKEHOLDER and BUSINESS FUNCTION is many-to-many. This is represented by "chicken feet" on both sides of the relationship line connecting these entities. This structure supports the stakeholder by business function matrix that is a fundamental part of several planning methodologies (see, e.g., [IBM, 1981]).

BUSINESS FUNCTIONs are typically hierarchically organized into sub-functions as indicated by the one-to-many recursive relationship on BUSINESS FUNCTION. Inventory Control, for example, is a BUSINESS FUNCTION that is a sub-function of the BUSINESS FUNCTION Operations.

During the process of IS planning, the STAKEHOLDERs analyze various ANALYSIS ASPECTs of a BUSINESS FUNCTION as prescribed by the planning methodology. For example, the planning methodology presented in [Dickson and Wetherbe, 1985] has five instances for the entity ANALYSIS ASPECT: problems, decisions, critical success factors (CSFs), efficiency, and effectiveness. These represent questions answered by the STAKEHOLDERs (e.g., what problems is the function having? what decisions are made in this functions? what are the critical success factors for this function?). Other possible ANALYSIS ASPECTS include: business transactions and activities, competitive strategy, and corporate objectives. Thus an IS planning methodology defines the population of the ANALYSIS ASPECT entity. All of the other entities are populated during the IS planning activity.

ANALYSIS ITEMs are the intersection between ANALYSIS ASPECTs and BUSINESS FUNCTIONs. That is, each BUSINESS FUNCTION has one or more ANALYSIS ITEMs for each ANALYSIS ASPECT defined in the IS methodology. If, for example, CSFs is an ANALYSIS ASPECT in the IS planning methodology, then the specific CSFs for each BUSINESS FUNCTION would be ANALYSIS ITEMs. Analyzing the Operations Function for CSFs would likely result in ANALYSIS ITEMs such as product quality and production technology. Hence, the population of the ANALYSIS ASPECT entity dictates what aspects of each BUSINESS FUNCTION to analyze (i.e., what questions to ask and what data to gather); it does not dictate how that data is gathered (techniques such as nominal groups, GDSS, JAD could be used).

The organization must establish what INFORMATION ELEMENTs are needed to for each ANALYSIS ITEM identified. What information, for example, is needed to identify and solve the problems, make the decisions, evaluate the CSFs that have been identified as ANALYSIS ITEMs? INFORMATION ELEMENTs such as **production mean** and **production variance** are likely needed to evaluate the ANALYSIS ITEM product quality (identified as a CSF). Information is produced by processing data. Hence, each INFORMATION ELEMENT is produced by one or more PROCESSes using one or more DATA OBJECTs. For example, results from quality control activities (e.g., sensor measurements extracted from a production process or sample testing measures) are DATA OBJECTs that could be used by a statistical PROCESS to produce the above INFORMATION ELEMENTs.

For planning purposes, DATA OBJECTs are organized into DATA CLASSes, each DATA OBJECT being assigned to one DATA CLASS. Hence, a DATA CLASS is simply a convenient name for a group of (presumably related) DATA OBJECTs. The organization of DATA OBJECTs into DATA CLASSes is an ad hoc process [Brancheau and Wetherbe, 1986]. It can result in ill-defined and possibly overlapping DATA CLASSes. Consider, for example, the DATA OBJECTs **number of stock-outs** and **re-order point**. These both describe the same conceptual "thing" in the world, an inventory item. However, without appropriate methods, these may be assigned to different DATA CLASSes (e.g., **number of stock-outs** in the Sales Orders DATA CLASS and **re-order point** in the Purchase Orders DATA CLASS). The conceptual representation presented below uses logical data modeling concepts to add structure to this process, yielding more useful and well defined DATA CLASSes.

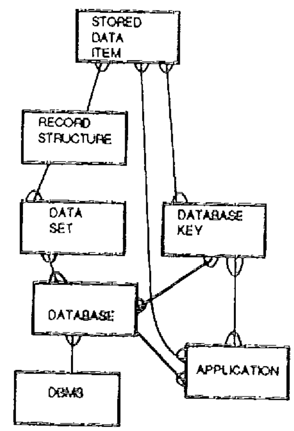
Stewardship for each DATA CLASS is assigned to one BUSINESS FUNCTION. Stewardship implies responsibility for the capture and maintenance of a DATA CLASS. A BUSINESS FUNCTION can have stewardship for zero or more DATA CLASSes. Stewardship is represented by the so labeled one-to-many relationship between BUSINESS FUNCTION and DATA CLASS. Similarly, the use of a DATA CLASS by a BUSINESS FUNCTION is represented by the so labeled many-to-many relationship between these entities. For example, while Operations and Marketing (instances of BUSINESS FUNCTION) might both use a DATA CLASS like product, only one of them can be its steward. The traditional information architecture matrix [Dickson and Wetherbe, 1985; Brancheau and Wetherbe, 1986] is represented in this data model.

The results of an information system planning activity represented in this data model provide a statement of information needs. The next step is to determine how effectively the current information systems meet those needs. Where there are gaps between important information needs and current capabilities, specific information system development efforts must be undertaken. The next section develops a model of existing information systems.

A Data Model of Implemented Information Systems

Implemented information systems are the vehicle through which information is made available to the organization [Carlis and March, 1984]. Figure 2 shows a data model of implemented information resources. Descriptions at this level reflect decisions made for operational efficiency. Seven entities are used: RECORD STRUCTURE, STORED DATA ITEM, DATA SET, DATABASE, DBMS, DATABASE KEY, and APPLICATION. This part of the IRM data model is populated during application development. It represents the data stored in a traditional data dictionary system (see, for example, [Allen, Loomis, and Mannino, 1982; Matthews and McGee, 1990]). This data defines the contents and capabilities of current information systems at the physical level. To effectively make use of this information (e.g., to choose which applications to develop and where it is appropriate to integrate new applications with existing ones), a conceptual level must be developed. The conceptual level is discussed in Section 3.3.

Figure 2
A Data Model of Implemented Information Resources



A RECORD STRUCTURE defines a physical file schema. Hence, a RECORD STRUCTURE contains some number of STORED DATA ITEMs (or fields). Although the same data value may be stored in different files, a STORED DATA ITEM is a part of only one RECORD STRUCTURE. An Inventory file, for example, may contain fields such as: item number, item description, price, bin location, re-order point, and quantity on hand. The value of item number may be stored in other files (e.g., in the Sales Order Line Item file as a foreign key representing the relationship between inventory item and sold item), however, it is a different STORED DATA ITEM in each of those files.

Each RECORD STRUCTURE has one or more DATA SETs, each defining a (possibly overlapping) set of data records that are stored in a single physical file. Each DATA SET has one RECORD STRUCTURE that defines its contents. Continuing the inventory example, one DATA SET could contain the descriptions of "fast moving" inventory items while another contains the descriptions of "slow moving" ones (both having the same RECORD STRUCTURE).

Each DATA SET is physically stored in one or more DATABASEs. Hence a DATABASE is defined by its schema (the RECORD STRUCTUREs of its constituents) and the instances of its schema (the DATA SET(s) of each of its RECORD STRUCTURES). Storing a DATA SET in more than one DATABASE means that the records in that DATA SET are duplicated (presumably for retrieval efficiency).

A DATABASE has some number of DATABASE KEYs. A DATABASE KEY is a set of STORED DATA ITEMS for which efficient access is supported (e.g., by index structures) by the database management system (DBMS) in which the DATABASE is implemented. A DATABASE can have many DATABASE KEYs and the same DATABASE KEY may be applicable to many DATABASES (provided they share RECORD STRUCTUREs). The same STORED DATA ITEM may be in many DATABASE KEYs. DATABASE KEYs are used by APPLICATIONs to retrieve and update STORED DATA ITEMS. An APPLICATION is a program that implements some required functionality. Design details such as screen and report layouts and algorithms are defined within the APPLICATION entity; however such details are beyond the scope of the current IRM data model (see, e.g., Matthews and McGee, 1990).

An Order Processing DATABASE, for example, could contain DATA SETS of fast and slow moving inventory items (having the same RECORD STRUCTURE), as well as customer, salesperson, and sales order DATA SETS (the salesperson DATA SET may also be redundantly stored in the Human Resources DATABASE). DATABASE KEYs could include: inventory item number, customer number, salesperson employee number, and sales order number. The Order Entry APPLICATION likely uses all of these DATABASE KEYs to validate customer orders.

This model describes the information resource at the physical level. It is at this level that data dictionary and database administration functions manage data. The concern is with specific implemented databases and the software that manipulates them.

This level of description is not sufficient, however, for effective IRM. Interrelationships among the various DATABASEs and among the APPLICATIONs are not specified. Many redundancies and inconsistencies are not identified. It is virtually impossible to map these implemented systems to the corporate planning models discussed above in any but the most cursory manner. To properly accomplish this mapping, a conceptual model must be developed. This conceptual model provides the key to understanding the implemented systems, their interrelationships, and their mapping to the information system plan.

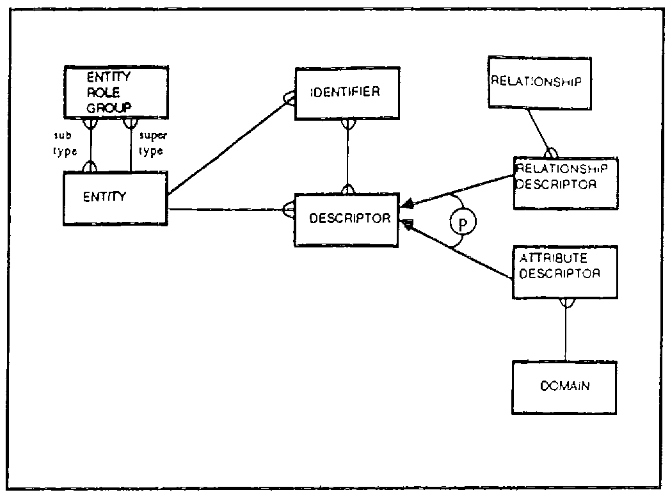
A Conceptual Model of the Information Resource

A conceptual model of the information resource must represent both the data content of the organization (a static data representation) and the ways in which this data is used (data dynamics) [ISO 1982]. These correspond to the major components of a semantic data model [Hull and King, 1987; Peckham and Maryanski, 1988]. They form the basis for automated database design tools and application generators [Carlis and March, 1984]. Each is discussed in the following subsections.

A Conceptual Model of Data Content

The conceptual model of the static data content of the information resource is a meta-data model (a data model of a data model [Carlis and march, 1984]). As shown in Figure 3, this it has eight entities: ENTITY, ENTITY ROLE GROUP, RELATIONSHIP, DESCRIPTOR, ATTRIBUTE DESCRIPTOR, RELATIONSHIP DESCRIPTOR, IDENTIFIER, and DOMAIN. This part of the data model is populated during system development. It is managed by data administration, who is responsible for maintaining its consistency and assuring that data needed by integrated applications is properly defined and captured. CASE tools provide the capability to capture at least some of this data.

Figure 3
A Meta-Data Model of the Static Data Content of the Information Resources



The (meta) entity ENTITY contains all entities represented in the information resource. An entity (or entity-type [Chen, 1976]) is any type (grouping, category) of thing (or event) about which information is required. Inventory item, customer, salesperson, employee, sales order, and sales order line item are examples of instances of this (meta) entity [March and Kim, 1988]. Each is an entity in the global corporate data model having RECORD STRUCTURE representations at the implementation level.

Each ENTITY has one or more DESCRIPTORs. A DESCRIPTOR is a fact that describes exactly one ENTITY. The entities ATTRIBUTE DESCRIPTOR and RELATIONSHIP DESCRIPTOR are subtypes of the entity DESCRIPTOR. That is, an ATTRIBUTE DESCRIPTOR is a

DESCRIPTOR and a RELATIONSHIP DESCRIPTOR is a DESCRIPTOR (indicated in Figure 3 by the arrowheads on the lines connecting these entities). Furthermore, a DESCRIPTOR must be either an ATTRIBUTE DESCRIPTOR or a RELATIONSHIP DESCRIPTOR but cannot be both. That is, these subtypes partition the entity DESCRIPTOR (specified by the circled p in the arc connecting these subtypes [Mark, 1983]).

Item number, **item description**, **price**, **bin location**, **re-order point**, and **quantity on hand** are examples of ATTRIBUTE DESCRIPTORs for the ENTITY **inventory item**. Each corresponds to one or more STORED DATA ITEMs in the implementation model (see subsection 3.4 below). Each ATTRIBUTE DESCRIPTOR has exactly one DOMAIN from which it draws. Multiple ATTRIBUTE DESCRIPTORS may draw from the same DOMAIN. The DOMAIN defines the set of allowed values for its associated ATTRIBUTE DESCRIPTORs. DOMAINS define input edit and validation constraints that must be included in the APPLICATIONs that manage the corresponding STORED DATA ITEMs.

A RELATIONSHIP DESCRIPTOR specifies how one ENTITY in a RELATIONSHIP describes the other ENTITY in the RELATIONSHIP (e.g., it specifies name of the descriptor and its the minimum and maximum degree). Without loss of generality, only binary RELATIONSHIPs are represented. That is, a RELATIONSHIP associates two ENTITIES. Each RELATIONSHIP DESCRIPTOR has exactly one RELATIONSHIP and each RELATIONSHIP has exactly two RELATIONSHIP DESCRIPTORs. Each of the two RELATIONSHIP DESCRIPTORs is a DESCRIPTOR related to a single ENTITY. These are the two ENTITIES related via this RELATIONSHIP.

For example, the RELATIONSHIP between the ENTITIES inventory item and sales order line item has two RELATIONSHIP DESCRIPTORs: **line-items-of-inventory-item** (a DESCRIPTOR of inventory item) and **inventory-item-of-line-item** (a DESCRIPTOR of sales order line item). **Line-items-of-inventory-item** specifies the set of sales order line items for a specific inventory item (its minimum degree is 0 and its maximum degree is "many"). This set describes that inventory item. Correspondingly, **inventory-item-of-line-item** specifies the inventory item ordered on a sales order line item (its minimum degree is 1 and its maximum degree is 1). One or both of these relationship descriptors may be physically implemented (i.e., have corresponding STORED DATA ITEMs at the physical level)

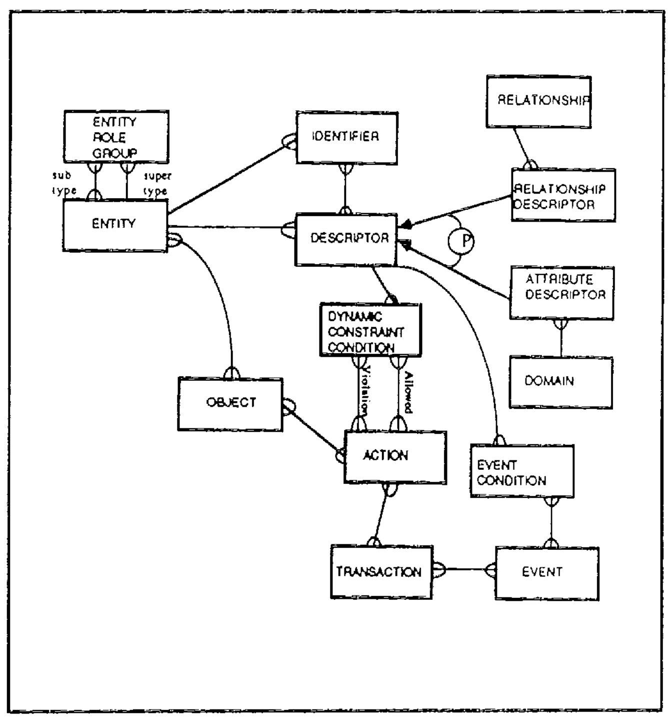
Each ENTITY has one or more IDENTIFIERs, each of which consists of one or more DESCRIPTORS. A single DESCRIPTOR may participate in zero or more IDENTIFIERs. For example, **inventory item number** is likely an identifier of the ENTITY inventory item.

Each ENTITY is a supertype of zero or more ENTITY ROLE GROUPs [Smith and Smith, 1977]. An ENTITY ROLE GROUP is a related set of subtypes or roles played by the supertype. The subtypes in the ENTITY ROLE GROUP are related by some constraint with respect to the supertype. For example, subtypes may partition the supertype (as described above), or they may be disjoint, but not completely cover the supertype [Mark, 1983]. Hence, an ENTITY ROLE GROUP has exactly one supertype ENTITY and one or more subtype ENTITIES. An ENTITY may be a subtype in zero or more ENTITY ROLE GROUPS. These facts are expressed in the many-to-one and many-to-many relationships labeled "supertype" and "subtype," respectively. For example, salesperson, secretary, manufacturing worker, and engineer form a disjoint ENTITY ROLE GROUP of the ENTITY employee (they do not overlap, but may not contain all employees).

A Conceptual Model of Data Use

Figure 4 augments the meta-data model of static data content with its dynamic aspects, that is, the ways in which it is used. Adding these dynamic aspects completes the conceptual data model of the information resource. Six entities are added: DYNAMIC CONSTRAINT CONDITION, ACTION, EVENT, TRANSACTION, EVENT CONDITION, and OBJECT. This data is captured during system development. There are significant deficiencies in the capability of CASE tools to represent system dynamics and to integrate them with the data description.

Figure 4
A Conceptual Data Model of the Information Resource



Each DESCRIPTOR has some number of DYNAMIC CONSTRAINT CONDITIONs. Each of these specifies the conditions under which specific ACTIONs can be (or must be) performed. Checking of DYNAMIC CONSTRAINT CONDITIONs may be immediate (e.g., as data values are changes) or delayed (e.g., after a transaction is completed). An ACTION updates or manipulates the values of the related DESCRIPTORs. DYNAMIC CONSTRAINT CONDITION has two different (many-to-many) relationships with ACTION: (1) ACTIONs that can be performed (i.e., are allowed) when the condition is satisfied and (2) ACTIONs that must be performed if the condition is violated (indicated by the so named relationships in Figure 4). An ACTION can be associated with many DYNAMIC CONSTRAINT CONDITIONs in either relationship.

For example, the DESCRIPTOR **employee marital status** may have the following ACTIONs: **change to single, change to married, change to divorced**, and **change to widowed**. A DYNAMIC CONSTRAINT CONDITION may state that if its value is married, then **change to divorced**, and **change to widowed** are allowed (but **change to single** is not). If an attempt is made to execute the ACTION **change to single**, then the constraint is violated and a constraint violation ACTION is executed (this ACTION could, of course, allow the update under certain conditions such as the correction of an error).

As in the Event-Transaction representation [De and Sen, 1984; Brodie and Ridjanovic, 1984; Olle, 1988], an EVENT is associated with a set of EVENT CONDITIONs. These specify the conditions (possibly involving related DESCRIPTORs) that define the occurrence of the related EVENT.

For example, the EVENT **Inventory Stock Out** occurs when the value of the DESCRIPTOR **quantity on hand** (of the ENTITY inventory item) goes to zero (DESCRIPTORs themselves do not have values; however, as will be seen later they are related to STORED DATA ITEMs which have values in their corresponding DATA SETS). Conversely, EVENT CONDITIONs may be unrelated to values of DESCRIPTORs but may be defined by conditions in the business environment. For example, the EVENT **Employee Termination** occurs when an employee leaves the company.

Each EVENT causes one or more TRANSACTIONS. The same TRANSACTION may be caused by more than one EVENT. Each TRANSACTION executes one or more ACTIONs, the results of which may satisfy additional EVENT CONDITIONs and thereby trigger additional EVENTs (TRANSACTIONs, and ACTIONs).

The **Inventory Stock Out** EVENT, for example, might cause the TRANSACTION **Place Emergency Re-order**. The ACTIONs for this event include: **Update Emergency Re-order Count, Select Vendor, Produce Purchase Order**, and **Notify Purchasing Manager**. **Update Emergency Re-Order Count** updates the value of the (STORED DATA ITEM corresponding to the) DESCRIPTOR **year-to-date emergency orders** (assuming it is allowed for the current conditions specified in the DYNAMIC CONSTRAINT CONDITIONs). The data may now satisfy EVENT CONDITIONs defining the EVENT **Improper Re-Ordering Policy**, which, in turn causes the TRANSACTIONs necessary to re-evaluate the re-order policy for that inventory item.

As in the Object Oriented paradigm [Peterson, 1987], an OBJECT is an arbitrarily complex organization of ENTITIES. Each OBJECT has a set of allowed ACTIONs (or messages to which it can respond [Dittrich, 1986]). The same ACTION can be performed on many different OBJECTS (polymorphism). An ACTION operates on the (STORED DATA ITEMs corresponding to) DESCRIPTORs of the OBJECT's ENTITIES as allowed by their DYNAMIC CONSTRAINT CONDITIONs.

For example, the OBJECT customer order may be composed of sales orders and their related sales order line items for a specific customer. It may have allowed ACTIONs such as: **total gross amount** and **insert line item. Total gross amount** calculates and totals line item extensions on an order.

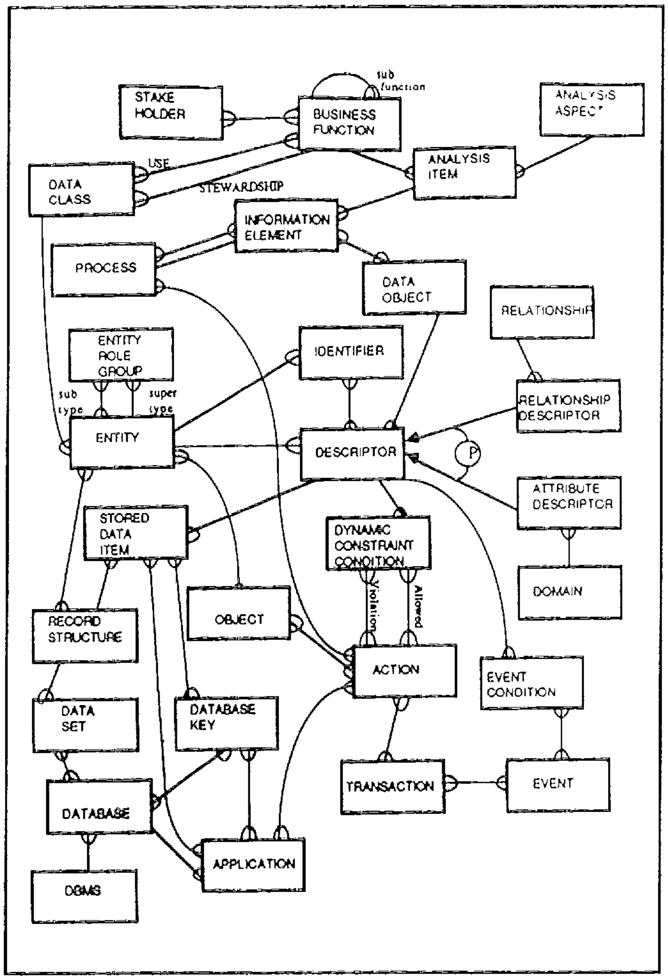
Once populated this data model contains a representation of the logical data maintained by the organization (i.e., it contains the corporate data model [Scheer, 1989]). In the next section, this corporate data model is mapped to the planning data model and the implementation data model.

An Integrated IRM Data Model

An integrated IRM data model is shown in Figure 5. In this figure, the conceptual model (Section 3.3) integrates the planning model (Section 3.1) with the implementation model (Section 3.2). Integrating with the planning model defines how the conceptual data supports the global information requirements of the organization. Integrating with the implementation model defines what parts of the plan are implemented and which are not. Furthermore, this mapping identifies conceptual data redundancies and opportunities for data sharing.

The planning model and the conceptual model should be mapped together when systems are developed. This mapping documents why the system is being developed. ENTITIES represent (part of) a DATA CLASS. DESCRIPTORs represent DATA OBJECTS. ACTIONs represent PROCESSes. Hence, a system is implemented to PROCESS (capture and manipulate) DATA CLASSes and DATA OBJECTs (data) and transform them into INFORMATION ELEMENTs needed by BUSINESS FUNCTIONs to evaluate various ANALYSIS ITEMs.

Figure 5
A Data Model for Information Resource Management



As discussed above, during the information planning process, DATA OBJECTs are often organized into DATA CLASSes in an ad hoc manner. This can result in ill-defined and overlapping DATA CLASSes. Using concepts from the conceptual level facilitates this process and improves the results. Each DATA OBJECT identified during the planning process corresponds to one or more DESCRIPTORs in the conceptual model. Each DESCRIPTOR describes exactly one ENTITY.

In this way, the ENTITY construct organizes DATA OBJECTs. Furthermore, ENTITIES are organized using the notion of subtypes and supertypes [Smith and Smith, 1977]. The highest level ENTITIES define major groupings of organizational data. These correspond to DATA CLASSes. Thus DATA CLASSes are defined using the same type of abstraction process that has been effectively used in the development of semantic data models [Brodie, 1984; Teorey, Yang, and Fry, 1986].

Consider the prior example where the DATA OBJECTs **number of stock-outs** and **re-order point** were assigned to different DATA CLASSes. Using the conceptual level, these are both DESCRIPTORs of the ENTITY inventory item. Hence they must be assigned to the same DATA CLASS (since each ENTITY is assigned to a single DATA CLASS). Hence, DATA CLASSes correspond to ENTITIES resulting in a more meaningful and well defined set of DATA CLASSes.

A PROCESS at the planning level is specified by one or more ACTIONs at the conceptual level. Each ACTION specifies one or more PROCESSes. A statistical analysis is an example of a PROCESS. Such a PROCESS can be specified by the types of statistics required resulting in a set of generic commands (ACTIONs) for a statistical package (an APPLICATION). PROCESSes can be considerably more complex, involving numerous ACTIONs. **Materials Requirements Planning** is an example of a PROCESS involving many ACTIONs (such as production planning and scheduling, raw material requirements determination, and production tracking).

Mapping from the conceptual level to the implementation level, each ENTITY is assigned to one or more RECORD STRUCTUREs. More than one ENTITY can be assigned to the same RECORD STRUCTURE allowing hierarchic files (with nested repeating groups) or complex objects in an object oriented DBMS. A RECORD STRUCTURE contains some number of STORED DATA ITEMs corresponding to the DESCRIPTORs of the ENTITIES assigned to it. Each STORED DATA ITEM corresponds to exactly one DESCRIPTOR. Each DESCRIPTOR has zero (if it is not physically stored) or more (if it is stored in multiple files) STORED DATA ITEMs.

For example, the ENTITY inventory item could be part of two different RECORD STRUCTUREs (file schemas): the first containing STORED DATA ITEMs (such as **item number, item description, price, bin location**, and **re-order point**) whose values are infrequently changed; the other containing more frequently changed STORED DATA ITEMs (such as **quantity on hand, quantity on order, quantity committed, month-to-date sales**). Update and backup and recovery operations can be very efficiently implemented using such schemas [March and Scudder, 1984].

ACTIONs at the conceptual level map to APPLICATIONs at the implementation level. An APPLICATION implements one or more ACTIONs. The same ACTION can be implemented by more then one APPLICATION. Each APPLICATION manages and retrieves data from an implemented DATABASE using some number of DATABASE KEYs.

This representation facilitates the location of (all copies of) all data maintained in organizational information systems. It shows which APPLICATIONs are responsible for maintaining which DATABASEs and provides a description of the characteristics of the data. In this way it is a support tool for

Data Administration, application development, and end-user computing.

The disparity between planned and actual information resource capability is represented either by the lack of conceptual description for existing planning elements or by the lack of implementation elements for existing conceptual descriptions. For example, the lack of PROCESSes and DATA OBJECTs for existing INFORMATION ELEMENTs indicates that the capability has not yet been planned. Lack of ENTITIES, DESCRIPTORs, and ACTIONs for corresponding DATA CLASSes, DATA OBJECTs, and PROCESSes indicate that the capability has not yet been designed. Lack of DATA SETs, DATABASEs, and their associated APPLICATIONs for the corresponding ENTITIES, DESCRIPTORs, and ACTIONs indicates that the capability has been designed but not yet implemented.

The development process is supported as planned capabilities are documented via DATA CLASSes and INFORMATION ELEMENTS (needed for the various ANALYSIS ITEMs) used to perform, manage, and evaluate various BUSINESS FUNCTIONS. It is tracked (for project management) as: (1) DATA OBJECTs and PROCESSes are identified from DATA CLASSes, BUSINESS FUNCTIONs, and INFORMATION ELEMENTs (analysis), (2) corresponding ENTITIES, DESCRIPTORs, and ACTIONs are defined (design), and (3) DATA SETs, DATABASEs, and their related APPLICATIONs are constructed (implementation). Integration is achieved through commonality of data description and control over data redundancy.

CONCLUSION AND DIRECTIONS FOR FURTHER RESEARCH

Despite the explosive increases in computing power and concomitant decreases in the cost of that computing power, effective management and utilization of organization information resources remains elusive. Neither the newest, most powerful computer hardware, nor the most current state-of-the-art database software can compensate for a failure to fully understand the nature of the organizational information resource. Without such an understanding and an explicit documentation of that understanding, effective Information Resource Management is not possible.

In this paper we have applied data modeling principles to the problem of understanding IRM data. We have identified basic entities about which data must be maintained if the information resource is to be effectively managed. This model includes the data required to plan, control, and develop the information resource. It defines three levels of IRM data: planning, conceptual, and implementation and specifies their interrelationships.

Future work will be aimed at fleshing out the model and integrating the model with the various methodologies used for information system planning and development. The issues to be addressed include:

(1) examining the breadth of information system planning methodologies, enterprise analysis methodologies, project management methodologies, systems analysis and design methodologies, software engineering and implementation methodologies, and system maintenance methodologies to add detail to the data model representation and to insure that it supports at least the basic features of these methodologies, and

(2) establishing a mechanism to integrate the results of these methodologies and associated tools to insure that planning efforts can be traced through to implementation, and do, in fact, impact actual system development efforts.

A second direction for future research is to exploit the model to facilitate and manage end-user computing. Issues involve the ability of end-users to interrogate a database containing the meta-data (an IRM database) and to use that data to enable access to the actual corporate data. Such access to corporate data must be controlled and managed. Use of an IRM database for this purpose will be explored.

Finally, establishment of an IRM program is not a trivial task. It implies a significant change in corporate culture from data ownership to data stewardship. It implies significant co-operative efforts on the part of data stewards and data users. It implies a significant change in the way in which system development efforts are funded and evaluated. Hence, there are a considerable organizational issues to be investigated. The key tasks are: (1) to establish the benefits of effective utilization of organizational information resources and (2) to assess the price an organization is willing (able) to pay in order to achieve these benefits.

REFERENCES

Allen, F. W., Loomis, M. E. S., and Mannino, M. V., "The Integrated Dictionary / Directory System," ACM Computing Surveys, Vol 14, No 2, June 1982, pp 245-286.

Brancheau, J. and Wetherbe, J., "Information Architecture: Methods and Practices," Information Processing and Management, Vol. 22, No. 6, 1986, pp. 453-463.

Brodie, M. and Ridjanovic, D., "On the Design and Specifications of Database Transactions," in: Brodie et al. (Eds.) On Conceptual Modelling, Springer-Verlag, New York, 1984, pp. 277-306.

Brodie, M., "On the Development of Data Models," in: Brodie et al. (Eds.) On Conceptual Modelling, Springer-Verlag, New York, 1984, pp. 19-47.

Carlis, J. V. and March, S. T., "A Descriptive Model of Physical Database Design Problems and Solutions," Proceedings International Conference on Data Engineering, IEEE, April 24-27, 1984, Los Angeles, CA.

Chen, P., "The Entity-Relationship Model - Toward a Unified View of Data," ACM Transactions on Database Systems, Vol. 1, NO. 1, March 1976, pp. 9-36.

Davis, G. B. and Olson, M. H., Management Information Systems: Conceptual Foundations, Structure, and Development, (2nd ed), McGraw-Hill, New York, 1985.

De, P. and Sen, A., "A New Methodology for Database Requirements Analysis," MIS Quarterly, Vol 8, No 3, September 1984, pp 179-193.

Dickson, G. W. and Wetherbe, J. C., The Management of Information Systems, McGraw-Hill, New York, 1985.

Diebold, J., "Information Resource Management: The Next Challenge," Infosystems, Vol. 26, No. 6, June 1979, pp. 50-88.

Dittrich, K., "Object-Oriented Database Systems: The Notion and the Issue," International Workshop on Object-Oriented Database Systems, Pacific Grove, California, Sept. 1986.

Dooley, R., "Linking Business and Information Systems Planning - Part I.," Spectrum, Vol. 3, No. 3, June 1986, pp. 1-6.

Dolk, D. R. and Kirsch, I. R. A., "A Relational Information Resource Dictionary System," Communications of the ACM, Vol 30, No 1, January 1987, pp 48-61.

Drucker, P., "The Coming of the New Organization," Harvard Business Review, January-February, 1988, pp 45-53.

Edelman, F., "The Management of Information Resources - A Challenge for American Business," MIS Quarterly, Vol 5, No 1, March 1981, pp 17-27.

Eftimie, R.-A. and Nikles, J.-C., "Information Architecture and Entity-Relationship Approach," in March (ed) Entity-Relationship Approach, Proceedings of the Sixth International Conference on Entity-Relationship Approach, New York, November 9-11, 1987, North-Holland, Amsterdam, 1988, pp 515-532.

Feuche, M., "AT&T Employs CASE Strategy for Competitive Edge," MIS Week, Vol. 11, No. 6, February 5, 1990.

Fong, E. N. and Goldfine, A. H., (eds) Information Resource Management - Making It Work, National Bureau of Standards Special Publication 500-139, Gaithersburg, MD, June 1986.

Fong, E. N. and Goldfine, A. H., (eds) Information Management Directions: The Integration Challenge, National Institute of Standards and Technology Special Publication 500-167, Gaithersburg, MD, June 1989.

Gartner Group, "Welcome to the Nineties: Its More Than Applications," Monthly Research Review, Gartner Group Industry Service, January 1990.

Goodhue, D. L., Kirsch, L. J., Quillard, J. A., and Wybo, M. D., "Strategic Data Planning: Lessons from the Field," MIS Quarterly, Vol 16, No 1, March 1992.

Hull, R. and King, R. "Semantic Database Modelling: Survey, Applications, and Research Issues," ACM Computing Surveys, Vol. 19, No. 3, Sept. 1987, pp. 201-260.

IBM Corporation, Business Systems Planning, IBM Manual #GE20-0527-3, July 1981.

ISO TC97/SC5/WG3, Concepts and Terminology for the Conceptual Schema and the Information Base, Van Griethuysen (ed), Report of ISO/TC97/SC5/WG3, March 1982.

Law, M. H., Guide to Information Resource Dictionary System Applications: General Concepts and Strategic System Planning, National Bureau of Standards Special Publication 500-152, Gaithersburg, MD, April 1988.

Lederer, A. L. and Mendelow, A. L., "Information Resource Planning: Overcoming Difficulties in Identifying Top Management Objectives," MIS Quarterly, Vol 11, No 3, September, 1987, pp 389-399.

March, S. and Kim, Y. "Information Resource Management: A Metadata Perspective," Journal of Management Information Systems, Vol 5, No 3, Winter 1988-89, pp 5-18.

March, S. T. and Scudder G. D., "On the Selection of Efficient Record Segmentations and Backup Strategies for Large Shared Databases," ACM Transactions on Database Systems, Vol. 9, No. 3, September 1984, pp. 409-438.

Mark, L., "What is the Binary Relationship Approach?," In Davis, Jajodia, Ng, and Yeh (eds), Entity-Relationship Approach to Software Engineering, Proceedings of the Third International Conference on Entity-Relationship Approach, Anaheim, CA, October 5-7, 1983, North-Holland, Amsterdam, 1983, pp 205-220.

Matthews, R. W. and McGee, W. C., "Data Modeling for Software Development," IBM Systems Journal, Vol 29, No 2, 1990, pp 228-235.

Mercurio, V. J., Meyers, B. F., Nisbet, A. M., and Radin, G., "AD / Cycle Strategy and Architecture," IBM Systems Journal, Vol 29, No 2, 1990, pp 170-188.

Navathe, S. and Kerschberg, L., "The Role of Data Dictionaries in Information Resource Management," Information and Management, Vol 10, 1986, pp 21-46.

Olle, T. W., Information Systems Methodologies: A Framework for Understanding, Prentice Hall, Englewood Cliffs, NJ, 1988.

Peckham, J. and Maryanski, F., "Semantic Data Models," ACM Computing Surveys, Vol. 20, No. 3, Sept. 1988, pp. 153-189.

Peterson, R., "Object-Oriented Data," IEEE Expert, March 1987, pp. 26-31.

Ross, R. G., Entity Modeling: Techniques and Application, Database Research Group, Inc., 1987.

Sagawa, J. M., "Repository Manager Technology," IBM Systems Journal, Vol 29, No 2, 1990, pp 209-227.

Scheer, A. W., Enterprise-Wide Data Modeling, Springer-Verlag, Berlin, 1989.

Selig, G. J., "Approaches to Strategic Planning for Information Resource Management (IRM) in Multinational Corporations," MIS Quarterly, Vol 6, No 2, June, 1982, pp 33-45.

Shank, M. E., Boynton, A. C., and Zmud, R. W., "Critical Success Factor Analysis as a Methodology for MIS Planning," MIS Quarterly, Vol 9, No 2, June 1985, pp 121-129.

Smith, J. and Smith, D., "Database Abstractions: Aggregation and Generalization," ACM Transactions on Database Systems, Vol. 2, No. 2, June 1977, pp. 105-133.

Synnott, W., The Information Weapon, John Wiley & Sons, New York, 1987.

Teorey, T. J., Yang, D., and Fry, J. P., "A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model," ACM Computing Surveys, Vol. 18, No. 2, June 1986.

Wetherbe, J. and Davis, G., "Developing a Long-Range Information Architecture," Proceedings of the AFIPS National Computer Conference, 1983, pp. 262-269.

Zachman, J. A., "A Framework for Information Systems Architecture," IBM Systems Journal, December 1987.