Design, development and implementation of a global information warehouse: a case study at IBM

Kathy A. Loeb*, Arun Rai†, Arkalgud Ramaprasad‡ & Srinarayan Sharma§

**Deloitte and Touche Consulting Group, †Department of Decision Sciences, College of Business Administration, Georgia State University, Atlanta, GA 30303, USA, ‡Pontikes Center for Management of Information, College of Business and Administration, Southern Illinois University at Carbondale, Carbondale, IL 62901-4627, USA, email: prasad@siu.edu [corresponding author], and §Decision & Information Sciences Department, School of Business Administration, Oakland University, Rochester, MI 48309-4401, USA*

Abstract. *The characteristics of a global information warehouse (GIW) can be understood with reference to the three dimensions implied by the three words in its name. These dimensions are boundary of the system, semiotic level of the objects in the system, and organization of objects in the system, corresponding to the terms 'global', 'information' and 'warehouse', respectively. This paper defines these three dimensions and describes the system characteristics that flow from the definitions. These characteristics also highlight the issues involved in the design, development and implementation of GIWs. The case study following the discussion of the three dimensions illustrates these issues.*

Keywords: Global information warehouse, information systems implementation, information systems design, information systems development, case study

A FRAMEWORK FOR ANALYSIS

The term 'global information warehouse' (GIW) is fairly new and is used quite loosely in the existing literature. According to Inmon & Osterfelt (1992, p. 76), 'an information warehouse is a central source of data, stocked with data extracted from different operational systems and standardized'. According to another view, huge amounts of useful information can be readily accessed by any user from a GIW with a consistent level of accuracy and control (I/S Analyzer, 1993). Some organizations such as Grand Metropolitan PLC consider their information warehouse essentially as a front-end to existing production systems, while others such as Owens-Corning Fiberglass Corp. view their GIW as a collection of subject- or application-oriented databases that are kept separate from the operational applications (Ambrosio, 1993).

In this paper GIWs are viewed as new hybrid information systems which bridge two broad classes of systems, namely storage and retrieval systems and processing systems. As storage and retrieval systems GIWs are like other systems in the class such as simple flat files, relational databases, distributed databases, integrated databases, knowledge-bases, local data marts and global data warehouses. (A data mart is defined as 'a subset of a data warehouse for a single department or function. A data mart may have tens of gigabytes of data rather than hundreds of gigabytes for the entire enterprise'. *TechEncyclopedia*, http://www.techweb.com/encyclopedia/.) As processing systems which transform lower level data objects into higher level information objects, GIWs resemble decision support systems (DSS), executive information systems (EIS), and on-line analytical processing (OLAP). Thus the objective of a GIW is not just the storage and retrieval of information objects, but also the programmed and ad hoc transformation of data to information for its defined user base.

The characteristics of a GIW can be understood with reference to the three dimensions implied by the three words in its name (Fig. 1). These dimensions are: boundary of the system, semiotic level of the objects in the system, and organization of objects in the system, corresponding to the terms 'global', 'information' and 'warehouse', respectively. Following is a definition of the three dimensions and a description of the system characteristics that flow from the definitions. These characteristics also highlight the issues involved in the design, development and implementation of GIWs. The case study following the discussion of the three dimensions illustrates these issues.

Boundary of the system

Definition

The logical boundary of an information system is defined by the location of the sources from which it obtains inputs, of users to whom it provides outputs, and of entities to which the

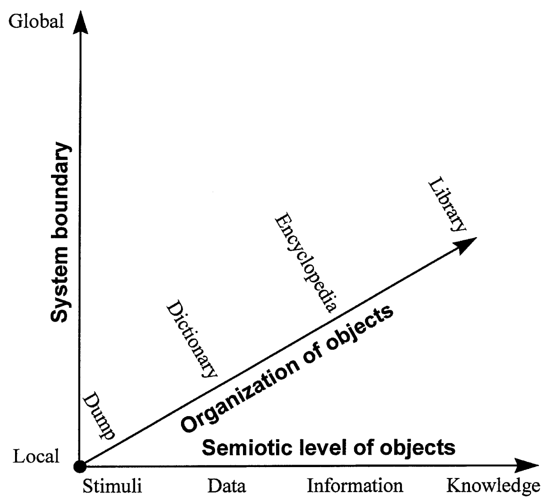

Figure 1. Global information warehouse in perspective.

information in the system pertains. Broadly speaking, based on its boundary, an information system may be characterized on a continuum from local through global. At the extremes of this scale, all three — sources, users and entities — are local or global; that is, local systems have local sources, local users and information about local entities, and global systems are the opposite. In reality, most systems tend to be in-between — partially local on some attributes and partially global on others. This is especially true with today's open systems which obtain inputs directly from the environment, provide access to those outside the organization, and contain information pertinent to entities within the organization as well as in its environment. Even though there may not be an instance of a purely 'local' or 'global' system, it is important to explicate the characteristics of these archetypes to understand and appreciate the differences between the design, development and implementation of systems that are hybrids of local and global characteristics.

Boundary-based system characteristics

We focus on the logical boundary of the information system and not on the physical configuration of the information technology that is part of the information system. Thus, on the local–global scale, personal databases are local, centralized databases are more global; data marts are more global than databases, but less global than data warehouses. Centralized, decentralized and distributed databases may be equally local or global depending upon their content. Following are some general characteristics of local and global systems.

Local systems are usually smaller in scale than global systems. They tend to have fewer sources of input, fewer users of its output, and information about fewer entities. Their scope too tends to be narrower, having fewer types of inputs, outputs and entities. Owing to the smaller scale and scope they tend to be less complex than global systems.

Local systems can be idiosyncratic or specialized without affecting the overall performance of other systems. They can be partially 'closed'. Their boundaries need not be permeable. On the other hand, components of a global system have to be standardized and general in order to be effective — or at least the components should be able to communicate with each other through interpreters and translators. In other words, global systems have to be 'open'. Their boundaries have to be permeable. For these reasons too, local systems tend to be less complex than global systems.

Local systems can be 'visible' in their entirety to a single observer; large parts of a global system are usually 'invisible' to any one observer. Naturally, it is easier for a person or group to obtain a synoptic perspective of a local system than of a global system. It is difficult to integrate the fragmented perspectives of a global system to obtain a similar synoptic perspective. A synoptic perspective helps ensure the overall efficiency and effectiveness of the system — and it is easier to do so with local systems. On the other hand, performance is likely to be suboptimal with a fragmented perspective — and that is more often the case with global systems. In the same vein, development of schemas for and learning about global systems is more difficult than for local systems.

It is for the above reasons that the process of design, development and implementation of a local system is different from that of a global system. The effort required to develop a global system differs both quantitatively and qualitatively from that required for a local system. Quantitatively, the scaling of effort from a local system to a global system will depend upon the nature of economies of scale and scope derived from globalization. Qualitatively, the effort required to develop a global system is of a different type, consisting of different skills and knowledge. These points are illustrated by the case study at IBM described below.

Semiotic level of objects

Definition

Broadly speaking, there are four semiotic levels of objects in an information system: morphological, syntactic, semantic and pragmatic.

> 'Stimuli [that can be perceived by the five senses] are at the morphological level . . . ; they are merely a collection of symbols, unconnected, without any meaning, and without any use. They are the raw materials for generating information. At the syntactic level, the stimuli are related to each other by rules to generate . . . what is usually called data. At the semantic level meaning is attributed to the stimuli and the relationships between them to generate what is often called information. At the pragmatic level the meaning of the stimuli is interpreted in a particular context; it is related to events and action, thus generating [what] may be called knowledge.' (Ramaprasad & Rai, 1996, p. 187)

The objects in a GIW are at the semantic level in the semiotic scale — one level above that in a database, and one below that in a knowledge-base. These objects may be graphs, tables or textual reports. As such they are more abstract than objects in a database and less than those in a knowledge-base. Creation and management of these objects requires an understanding of their semiotics.

Semiotic level-based system characteristics

Many factors affect the progression of objects from one level to another along the semiotic scale. Each step, namely the transformation of stimuli to data, data to information, and information to knowledge, may be unifinal, multifinal or equifinal. A unifinal transformation is one where there is a unique mapping from one level to another independent of the cognitive agent. A multifinal transformation is one where different outputs (data, information or knowledge) can result from the same inputs (stimuli, data or information, respectively). An equifinal transformation is one where the same output results from different inputs. Unifinal transformations are cognitive-agent and context independent. On the other hand, equifinal and multifinal transformations are cognitive-agent and context dependent. The above issues are discussed in greater detail in Ramaprasad & Rai (1996).

Each upward semiotic transformation therefore introduces new complexity and variation in the output objects. Thus the complexity of and variation in data is different from that of stimuli;

the complexity of and variation in information is different from that of data; and the complexity of and variation in knowledge is different from that of information. It is for these reasons that the process of design, development and implementation of a knowledge system is different from that of an information system, and of an information system is different from that of a data system. At each level managing the complexity and variation requires different skills and processes. The skills and processes required at the information level are illustrated by the case study at IBM described below.

Organization of the objects

Definition

Books in a library are organized according to the Dewey classification system or the Library of Congress classification system. These are widely accepted systems which not only help librarians to organize the books, but also help users to find the right books. Without these classification systems libraries would be chaotic. Today, these classification systems have become so institutionalized that almost all books are given a classification code at the time of publication itself.

Similarly, data-, information- and knowledge-objects have to be organized using a classification system. This classification system will affect the efficiency and effectiveness of the objects' retrieval. Whereas the generation of these objects will not be affected by their subsequent organization, their dissipation will be adversely affected (Ramaprasad & Rai, 1996). The consequent imbalance between generation and dissipation will create the classic symptoms of 'information overload' and 'information in jail' (Orr, 1991). The objective of organizing the objects is to maintain their generation and dissipation in balance.

Organization of object-based system characteristics

The objects may be organized using a simple or a complex classification system. According to the law of requisite variety the number of categories in the scheme has to match the variety of objects. A system with more categories than required by the variety of objects will be redundant; one with fewer categories than required by the variety of objects will be ineffective. Furthermore, how these categories are derived will determine the complexity of the system. As with any taxonomy, if the categories are derived from a few underlying dimensions then the classification system will be simple; if there are many underlying dimensions then the classification system will be complex. Not surprisingly, the trade-off between the efficiency of having only a few categories and the effectiveness of having many categories plays an important role in determining the actual level of complexity (or simplicity) of the classification system.

Naturally, objects at different semiotic levels have different attributes and will require their own classification systems. Just as the same classification system cannot be used to organize a dictionary and an encyclopaedia, a data dictionary cannot be used to organize a set of tabular data and graphs in an information-base, or a set of rules in a knowledge-base. Each classification system has to correspond to the attributes of the objects at that semiotic level. Furthermore, these systems will have to match the increase in the complexity and variety of objects as the objects progress along the semiotic scale, as the system's boundary expands, and as the system itself evolves.

There are no well-defined anchors representing progression along the 'organization of objects' dimension. Based on our earlier analogy, we will designate a 'library' organization as the highest form. Such an organization will require a theory of the objects. There are no current examples of organizational storage and retrieval systems with such an organization. One step below 'library' form is the 'encyclopaedia'. In an encyclopaedia, objects are organized by subject and then ordinally on an attribute — generally alphabetically or temporally. Warehouses and marts approximate such an organization. A step below the 'library' form is the dictionary. Here the objects are simply organized ordinally on an attribute — again generally alphabetically or temporally. Databases and knowledge-bases approximate such an organization. The lowest form of an organization is called a 'dump'. It connotes no organization. Objects are stored with little consideration for their effective and efficient retrieval.

Unfortunately, there is as yet no widely accepted classification system for organizing information objects in corporations and other similar organizations. A customized system has to be designed, developed and implemented along with the storage and retrieval system by those that usually have no training either in library science or in epistemology (the theory of knowledge). The following case study illustrates some of the issues entailed in the design, development and implementation of a system for organizing information objects.

Interaction of boundary, semiotic level and organization

An information system may grow or evolve along any one of the three dimensions discussed above without evolving on the others. Thus, for example, it may expand its boundary without changing the semiotic level or the organization of objects: a local database may be expanded into a global database without changing the underlying structure. Such changes are easier to manage than changes along two or more dimensions. However, evolving information systems along all three dimensions is increasingly essential to support information needs of knowledge workers.

When changes occur along two or more dimensions simultaneously, the complexity of design, development and implementation is compounded. Thus it is very much easier to move from a global data warehouse to a global information warehouse than it is to move from a local database to a global information warehouse. Some of the problems associated with changes along multiple dimensions are also illustrated in the following case study.

DATA ACCESSIBILITY AT IBM: A CASE STUDY

The division of IBM in review (henceforth, we will use IBM to refer to this IBM division) has branches that are geographically located in many cities in the USA and many countries around the world. The availability of timely, accurate and relevant information is a necessity for conducting the business efficiently and effectively around the globe. In 1987, however, it was imperative to be able to find answers to business queries such as the following: What is the revenue per customer for a particular set of products? What is the top quality problem for a set of products during a specified time period? What is the defect per line of code for a given software product?

IBM was seeing leaders in the use of information technology able to tell the exact location of their different products and their inventory as a business advantage, rather than a number of people spending an incredible amount of energy and time finding information when they could be analysing it. It was not that IBM did not have relevant data. In fact in the words of the project leader, ‘We were data rich but information poor’. The data were (and in many cases still are) stored on different (legacy) operational databases such as finance, manufacturing, personnel and inventory. However, the data in these databases were in different form and format and incompatible with one another. Consequently, the customers (internal users within IBM) faced data accessibility problems.

Clearly an enabling technology was needed to solve the data accessibility problems. Since information technology (IT) enables coordination across space (geography) and time (Scott Morton, 1991), it made sense to use IT to logically integrate data spread all over the world. With this spirit, a blueprint was laid out at IBM to build a global information warehouse under the auspices of an information deployment mission developed to solve the data accessibility problems. The objective was that customers’ (internal users) prioritized data/information needs be understood and their ability to access the required data/information be improved in a timely and automated manner. This goal was primarily accomplished through the development and implementation of a data warehouse and an information warehouse.

Architecture of information deployment project

The information deployment project was envisioned to consist of a comprehensive relational database that would extract data from a number of internal and external databases and eventually constitute the data warehouse. Several front-end tools would then be hooked onto it to extract the desired information through a number of self-contained queries to answer specific problems of specific customers. This information would reside in the information warehouse.

Figure 2 shows the information deployment architecture. At the top of the figure are IBM’s internal customers of information — management, knowledge workers and executives. While the definitions of management and executive customers are self-evident, knowledge workers are those who have a justified need for information to help them do their job. Thus nearly anyone within IBM can be regarded as a knowledge worker.

These internal customers required information in two different ways. A small number of customers, who were technically competent, did not mind hooking onto the relational database themselves. They would define their own application, or tailor someone else’s ‘canned’ applications to their own specific needs. The other, larger set of customers required that information be delivered to their electronic in-baskets and would never directly manipulate data

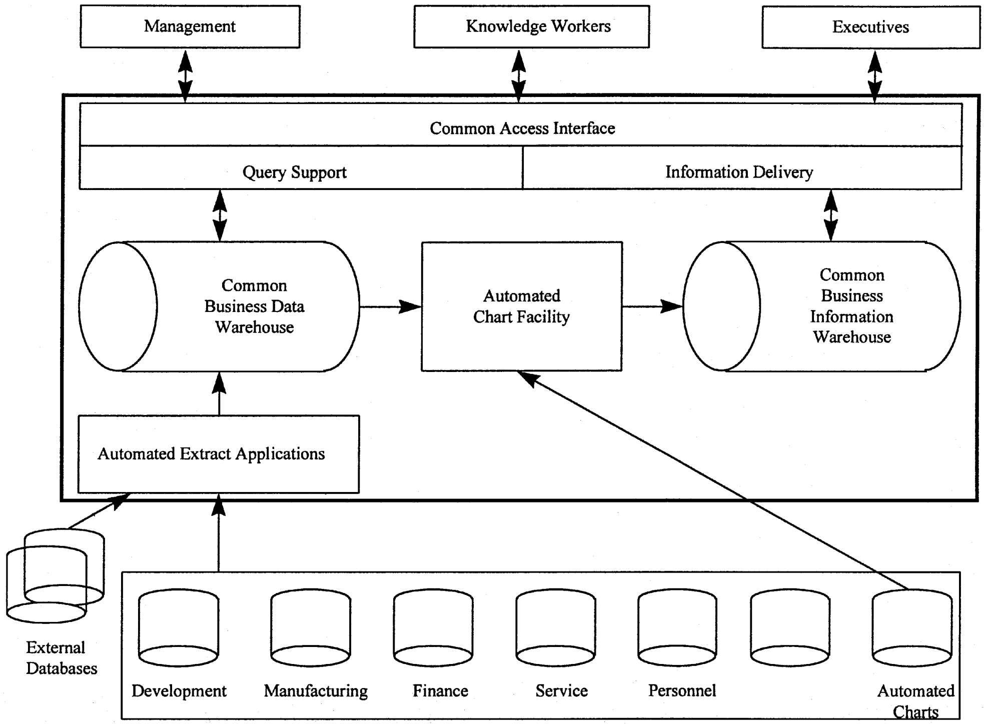

Figure 2. Information deployment architecture.

stored in the relational database. By the time the project was finished, there were about 200 direct users and about 12 000 indirect users.

The Common Business Data Warehouse (which is conceptually the same as a global data warehouse) stores data from different internal and external databases to fulfil customers' data requirements. (It is important to note that these internal databases can be located in any branch of the division. Similarly, the external databases can be located anywhere.) Most of the data stored in the data warehouse is automatically extracted from various internal operational (such as development, manufacturing, finance, service, personnel, customer support) as well as external databases through automated extract applications. These applications continually extract data from the databases to update the data in the data warehouse.

The Automated Chart Facility links the data warehouse and the information warehouse. This facility automatically generates many levels of information, such as specific monthly, weekly or even daily charts and reports and stores the same in the information warehouse. Thus the information warehouse can be viewed as a collection of relatively static charts and reports in a library, configured to store different levels of abstraction which in turn can be used by end-users to find answers to ad hoc queries.

The customers can access data from the data warehouse and charts and reports from the information warehouse through a 'Common Access Interface'. This interface has a base of tools for queries and analyses. We categorize these tools into 'Query Support' tools used to access and manipulate data from the data warehouse and 'Information Delivery' tools used to access reports and charts in the information warehouse. (IBM's query management facility connected to a relational database (DB2 MVS) was used as the primary query support tool.) The common access interface in conjunction with the query support tools drives off a DB2 relational database and facilitates the ad hoc queries of the customers. An IBM tool called 'executive decisions' was used as the primary 'information delivery' vehicle.

As evident from the above discussion, the common access interface is a very important component of the architecture. It shields customers from technical details of the system and provides flexible access to data and automatically generates charts and reports. The query support and information delivery tool sets help access data and information and transform data into information which makes it possible to make proactive decisions — decisions that help in addressing a 'situation' before it turns into a 'problem'.

Development process

The information deployment project at IBM evolved incrementally. Initially one subject area was approached at a time. The project team first focused on financial information. Within this area, it first focused on revenue data and then on expense data. The consolidated data and information from a subject area were immediately made available to the users. Once the (direct) users gained some experience, they started building the tables themselves and asking for more data. For each direct user there were a number of downstream (indirect) users who were receiving benefits. It was a 'domino effect' that led to the creation of more and more tables and consequently more and more demand. At the same time, the team members also gained experience, which helped them to work in more than one area simultaneously. In the first year, two to three different subjects were put up. The number of direct users was about 55. By the end of the second year, the number of areas was almost twice that in the first year. Over the course of four years, the number of direct users reached 200.

For all the areas, the development process consisted of three stages: requirements determination, design and implementation (coding and testing).

Requirements determination

A project team approach was used to elicit customers' data and information needs. The project teams were constructed from three constituents: end-users, system analysts and 'intermediaries'. End-users were the internal customers who needed information to perform their daily tasks. System analysts were the information systems (IS) professionals who had had some years of experience in handling the organization's information needs. As the name indicates, these IS personnel were less programmers, more analysts. 'Intermediaries' were the individuals who had the skills to understand the end-users' needs and translate them into a language that was comprehensible to the system analysts. The composition of the team is congruous with the principle of cross-functional integration emphasized by the Quality Function Deployment approach to ensure quality of deliverable products (Zultner, 1993).

The customers were first taken through a ‘wish list meeting’ where all the key representatives specified their data and information needs. They were asked to bring to the meeting all the reports and analyses, not only the ones they were using at that time. The process was exhaustive as the customers were made to articulate what they wanted to do, how they measured progress and quality of the task, what information they already had, and what other relevant information they might need. Customers were also asked to share samples of measurement(s) they had been using. From this process, the most elementary (the term ‘lowest common denominator’ was used by the project leader) data elements for all possible customers were identified across all functional areas. Then these data elements were combined with other data elements, restructured, and repackaged into information which the project team thought the customers needed. The packaged information was presented to the customers and feedback was obtained. Every time some new requirements were identified they were communicated back to the stakeholders from whom they were obtained to check their correctness and accuracy. The process was repeated until the customers were reasonably satisfied.

Often the project teams found it easier to work backwards from the users’ requirement determination to produce the information users wanted. They would start with the users requirements, such as certain charts and reports, and work backwards to find necessary primary data to construct those charts and reports.

The whole requirements determination process required a very strong ‘partnership’ among end-users, intermediaries and system analysts. Customers needed to understand the importance of communicating clearly with system analysts. However, they found it difficult at times as they were not well versed in the terminology and metaphors used by information systems personnel. Since system analysts designed the system, it was critical that they really understood the needs of end-users. However, analysts were handicapped in the same way end-users were — they were not well versed in day-to-day business terminology and metaphors. The communication between end-users and system analysts/programmers was facilitated by the intermediaries. These individuals understood the languages of both the end-users and system analysts and had special skills, which are discussed later under ‘Implementation challenges’. (The project leader belonged to the ‘intermediaries’ category.) Synergism was needed among the three constituents to effectively elicit the information needs of end-users, which developed only over a period of time. The end-users improved over time in their ability to articulate their needs as they started to understand what they were being asked for. Similarly, the intermediaries became better translators, and IS professionals became better interpreters. It is important to note that there was some lead time before intermediaries and IS personnel were able to gain the customers’ support and confidence. Because they were not a part of the business the customers were, they had to learn about their customers’ business before they were able to ask the right questions. It is also important to note that intermediaries and IS personnel always interacted with the customers together to make sure that they heard the same thing. The intermediaries also worked with the same IS personnel from one subject area to another; this made it possible to develop better understanding, trust and confidence between them. It also helped to work at a faster pace and facilitated better design.

There were four intermediaries throughout the project. The number of IS personnel varied between 10 and 12. Some of the IS personnel were consultants. At any time an intermediary had at least one team to work with; however, they could be on three teams at the same time. The IS personnel did not work independently, but were assigned to the intermediaries who always worked in different subject areas.

Design: mapping the elicited data and information to business processes

Once the data and information requirements identification was complete, they were arranged into separate tables according to subject areas. Hundreds of queries were then formulated against these tables to make it possible to answer specific questions from any customer from any business area. (Some of these queries were later used as a starter set of questions for user training.) The information deployment model was used to cross-map the data and information categories, decision and query support tools, and respective business processes and business segments (see Fig. 3). (A business segment is defined as any functional area of IBM such as marketing, finance, manufacturing, etc. A business processes is any process used by any business segment. A business segment owner may own multiple business processes such as software and hardware development.) This mapping was done to ensure the customers do get the right data and information. (The internal customers were in a sense the owners of business processes and segments.)

So far we have discussed the information requirements determination process and the mapping of the same to relevant business processes. It was essential to prioritize these requirements for implementation. Priorities were established based on the importance of information for organizational goals and objectives. A continuous dialogue with senior management, who were more knowledgeable about such needs, enabled making this judgement. As a rule-of-thumb, the 80–20 rule was followed: priority was given to those 20% of the requirements that would result in 80% of value to the customers.

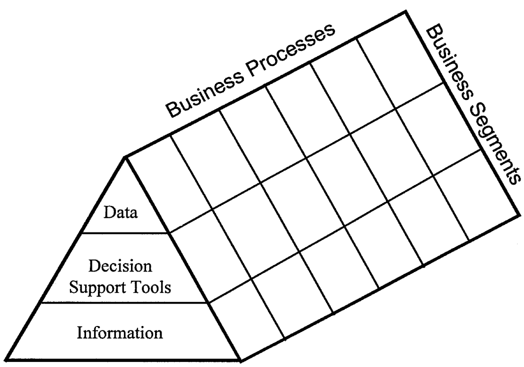

Figure 3. Mapping of information.

Implementation: coding and testing

Once the requirements identification and design phases were complete and the priorities had been established, the DB2 MVS relational database was physically created and its link to external and internal databases through the automatic extract facility and to the information warehouse through the automatic chart facility was established. IBM's query management (for query support) and executive decisions (for information delivery) tools were hooked onto the data warehouse and information warehouse, respectively. A friendly interface was created to provide the user with a common interface for both query support and information delivery. After an initial testing and debugging, the system was made available to the customers in the relevant subject areas for further testing and debugging.

The testing period, appropriately called 'customer expectancy testing', lasted from 2 to 4 weeks depending on the size of the project, the number of tables customers were using, their schedule, etc. They were asked for their assessment in terms of meeting their original requirements. Their inputs were taken and were incorporated until their requirements were reasonably (about 80%) met.

Implementation challenges

Despite the proliferation of information technology in organizations, the implementation of information systems remains a significant issue (Lucas *et al.*, 1990). Successful implementation of modern technology, innovations and management is crucial for enhancing the productivity and competitive position of an organization. However, the successful implementation of IS in organizations remains a challenge. The problems involved are accentuated by numerous past IS implementation failures (Lucas, 1981; Zmud, 1983). Implementation of the GIW at IBM was no exception and posed many challenges. In the following paragraphs, we discuss these challenges.

Financial and political support for the project

Both financial and political support were needed for the project. Financial support was easier to obtain. The division president had been closely tied with IBM customers in similar activities; he understood the implications of the information deployment project and always supported generously. The project never overran its budget. In fact, it usually had some money left at the end of each financial year.

Politically, it was much more difficult to gain the support for the project for many reasons. First, it was difficult to show the savings in tangible terms. Hence the project was always portrayed in terms of cost avoidance. At the end of each one of these major subject area activities, customers would be asked to estimate the time the new system was saving them in comparison to the old. A model was developed to estimate productivity saving. The time saved was multiplied by the person's wage to find out total savings. These savings were showed to the customers as cost avoidance for that subject area. Over a period of time, the project teams would show over the chart by subject areas the dollars that IBM did not have to expend because of productivity improvement. This saving was compared with the entire development cost which included the cost for requirement analysis, design, and coding and testing and presented to the customer.

Secondly, political support was further imperilled by the substantial time lag before the productivity improvement could be shown. The average lead time was 6–9 months for the entire project. Some of the subject areas took longer depending on the complexity of the data and difficulty of getting them from other locations. (Getting data from other locations in itself proved problematic, particularly if it was managed by another division. The project team had to negotiate with other divisions. Sometimes it had to pay the expenses incurred to the other division in formatting and sending those data. Other times the project would send its own IS personnel to work on the other division's data.) For a long time people have seen investment in IT as an expense. Because the information deployment mission was of part of the IT budget, the long lead time reinforced this view. It took four years before productivity enhancement could be made visible in all the subject areas. In the absence of quick results, the customers could argue that the old system worked better and the money spent on the information deployment mission could be better spent on product development.

Third and politically most sensitive, the project team had the formidable task of dispelling the apprehension of the customers that the productivity improvement would result in a loss of personnel working for them. If the project team were to tell the customers that using the system would save them 20% of the time and the labour (people) cost, customers would interpret this as meaning that they would lose 20% of the head count. So the project team was very careful not to talk about benefits in terms of reduction in the number of personnel. It always talked in terms of money and time customers didn't have to expend additionally as well as the ability to do increased analysis with the same people. There was no direct layoff because of the project.

In spite of these sensitivities by the project team, the customers would come back and ask the inevitable question of reduction in head count. The project group had to circumvent these questions arguing that it was not directly part of the development process. It was a support group that was funded to work on the information deployment project. Politically, the project group would make the case that it was not asking customers to pay for this support. This was a value-added service for them to run their business and they could use the saved money and people time to do other things that they could not do before.

Enterprise culture

An organizational culture that takes advantage of the trust and respect of the users for integrity and professionalism is likely to implement and benefit from a system like GIW. Mutual trust among executives, management personnel and knowledge workers was found a necessity and had to be nurtured over a period of time. In the words of the project leader, 'changing cultural aspects was the most difficult of all the challenges'. In the following paragraphs, we discuss different aspects of the enterprise culture.

During GIW development the most difficult barrier to overcome was the ownership of data. It was the political structure and organizational culture that made people feel that information was something to be carefully analysed and evaluated, not necessarily an asset that needed to be shared. From their perspective, the GIW system was taking what they always owned and giving it to 'everyone'. It was very difficult to make the case that sharing the information generated may raise questions and their ability to answer those questions made them more valuable. If information is power then the understanding of information (analysis) is even more power.

There was also a sense of insecurity among 'owners' of data. The insecurity arose out of the system's ability to automatically transform raw data into different levels of information and making them available to all the levels of management at the same time. Owners did not have a chance to check its meaningfulness and accuracy. This induced fear in owners — fear of not being able to correct their data before it was too late. Driving away this fear and insecurity proved to be a major step in securing customers' cooperation.

The sense of fear and insecurity was exacerbated by a potential use of information as a weapon rather than a tool. While information obtained from the GIW system could be used proactively to solve business problems, it could also be used to reprove or punish the owners of information if the information was not so desirable. There was the threat that some mangers would pick up the phone and 'beat up' the person who 'owned' the 'undesirable information' (problem). Unfortunately, the owners of information were observed not to share 'their' information if it had been negatively used (as a 'weapon') against them. The potential negative use of information reinforced a sense of fear in owners. Overcoming these problems required further developing trust between owners and users, and a commitment on the part of the users. Such trust and change in attitude increased over time. In addition, executives accessed information directly from the system and removed the traditional organizational hierarchies and subordinates for 'filtered' information delivery.

Skills

As is the case with any information system, people with special skills were needed to develop, implement and use a GIW. These skills usually included ability to use decision-support tools, query management facilities and data interpretation systems, and perform 'what-if' analyses. People with these skills are usually hard to find. Fortunately, IBM had a number of people who possessed these skills as they had been in positions requiring them. (Because personal and host (mainframe) AS systems have been primarily used for 'what-if' analyses and other decision-support and statistical analyses, and because these systems have been with IBM for a long time, there was a population that was very skilled in using these systems.) It takes a certain desire to learn new and additional technology skills. Attitude and cultural issues were found to be more static than the ability to learn technical skills.

The need for individuals with the abilities to perform 'what-if' analyses and use decision-support tools notwithstanding, another type of individual was also required. These individuals not only understood the way the business was run, but also had the technical acumen to understand different decision-support tools and apply them to customers' business. These individuals had a high ability to work and communicate with the IS group. Unfortunately, the individuals who possessed these skills were limited in number; IBM had to develop additional ‘business analyst type’ skills in individuals from scratch.

User training

Technical training was imparted to the users by providing them with a ‘starter set of charts and measurements’ to set them up with tools and techniques so that they could use the GIW. As mentioned earlier in the development process, the starter set of questions usually included half a dozen charts and reports that the customers brought to the ‘wish list meeting’ and the queries that were initially created against different tables in the database to obtain those charts and reports. The expense on the user training was minimal as people typically were doing information analysis of some sort as part of their primary job responsibility. Some of them already had some of the technical skills needed to use the system. Also, the user-training period was very short.

However, it was difficult to train users to be proactive. After all, the users had analysed information lifelong after the fact. Turning them to be proactive from being responsive clearly called for a shift in attitude and mental maps.

Organizational impacts and benefits

IBM’s information deployment mission placed an extraordinary emphasis on the availability of timely and relevant information to end-users. The organizational efficiency and effectiveness resulting from data integration have resulted in a higher quality of work and better business decisions by the end-users. These advantages may further manifest into better image and higher profit for the organization. Below we describe some of the benefits of the GIW system.

The consolidation of data in one place has made it possible for end-users to get consistent, accurate and reliable information in a timely manner. For example, earlier it was difficult to make sure that IBM had enough of a particular product to fulfil certain customer orders. However, the current system has improved inventory management by an order of magnitude. This has allowed knowledge workers to spend more time analysing information rather than finding it and thus has increased their productivity. As mentioned before, the increase in productivity has been between 20 and 30% on average. This has also helped users to make ‘informed’ business decisions and thus be more effective in carrying out their jobs. As the warehouse data draws on both internal and external databases, it has provided IBM the ability to quickly and effectively respond to changes in the marketplace.

Because the data being used for extracting information reside on different operational systems with varying definitions, it also shields customers from the intricacies and complexities of fragmented and dispersed operational databases. Furthermore, use of a consistent base of menus and tools has resulted in less cognitive burden on the end-users.

The information warehouse has also contributed to avoidance of many expenses which would have been necessary if information had to be kept on different databases. Getting the desired information in such cases would also take time and effort. Thus many costs have been avoided by saving time and reducing the expenditure on the operation and maintenance of disparate databases.

Because operational systems were built to run day-to-day business, data residing in these systems do not lend themselves to trend analysis. However, the data warehouse stores both present and past data and can support various decision-support applications including trend analysis. This enhanced decision support has helped IBM to act proactively and address a 'situation' long before it becomes a 'problem'. For example, in the past it was difficult to isolate customers who had little or no quality problems with IBM products because of their skilled support structure from those customers who always had problems because they did not have the requisite support structure in place. It was difficult to see a pattern across the product lines. However, with the help of the new system it was easy to find the pattern that some customers were always in the critical situation with problems. These customers could now be helped before they came up with problems.

The project leader of the information deployment team aptly summarized the benefits of the system as:

> 'We really have transformed knowledge workers from chart makers into, what they are paid for to do, thinking, analytical, proactive, decision makers.'

A RETROSPECTIVE CASE ANALYSIS

The above case study provides a historical description of the design, development and implementation of a GIW at IBM and can be used for answering the following questions that are based on the framework described earlier in this paper. These answers will help generalize the insights from the IBM experience to other settings. The questions are:

- How did IBM manage the progression towards a global system from a conglomeration of local systems? What strategies did they use? What alternative strategies could they have used?
- How did IBM manage the movement from data to information along the semiotic scale? What strategies did they use? What alternative strategies could they have used?
- How did IBM manage the progression from database to warehouse in the organization of objects? What strategies did they use? What alternative strategies could they have used?
- What contextual factors were important in IBM's successful implementation of a GIW?

The fact that IBM adopted particular strategies in successfully designing, developing and implementing its GIW suggests neither that these are the only strategies available, nor that they will always be successful. One cannot even judge whether they were right or wrong. However, IBM's success clearly provides evidence that these are viable strategies. The case study provides a reference point for mimicking the strategies, refining them, modifying them, or rejecting them in other similar contexts.

Strategies for globalization

Obtaining global answers to traditional queries about business operations was a primary driver of the global information warehouse development at IBM. At the same time, obtaining global answers to non-traditional queries, which hitherto could not be posed because of systemic constraints, was another primary driver. Thus the new system would not only permit, but also encourage, global exploration and enquiry. This global character is reflected in the variety of internal and external databases the team sought to integrate, as shown in the bottom layer of Fig. 2. They sought to integrate development, manufacturing, finance, service, personnel and external databases. It is also indicated by the fact that the system was designed to cut across multiple business segments and business processes, as shown in Fig. 3.

The number of direct and indirect users provides another indication of the nature and extent of globalization. In the first year there were about 55 direct users, in the second about 100, and by the end of the fourth year about 200. By the end of the fourth year, there were about 12 000 indirect users. Taken together, the rate of growth of direct and indirect users can only be characterized as exponential. However, the globalization effort in terms of resources expended did not increase even linearly, let alone exponentially — it remained more or less constant throughout the project. IBM appears to have derived economies of both scale and scope in its implementation of a global information warehouse.

Furthermore, a core team of intermediaries and IS personnel worked throughout the entire project. This common thread may have played a key role in ensuring that the GIW was a truly integrated system, and not simply an agglomeration of local systems. In a sense it could be argued that this core team was able to evolve the logical vision as well as the physical implementation of the GIW in a coherent and systematic manner.

Thus IBM used an incremental strategy in the context of a synoptic architecture for globalization. However, it appears to have emphasized the incremental strategy considerably more than the synoptic one. The initial synoptic architecture (Fig. 2) — which does not appear to have been modified subsequently — provided guidance and direction to the ongoing interaction and iteration of ideas between the design teams and the customers. Different functional areas were included in the process sequentially and a domino effect was utilized in generating the information tables. This incremental strategy facilitated the evolution of the details of the GIW in such a way that they fit the overall architecture as well as the needs of the customers; the needs too evolved with the design, development and implementation of the GIW. The incremental strategy may have been time-consuming and people intensive, but it appears to have been effective. Thus IBM appears to have succeeded by taking a combination of the high road and the low road (Allen & Boynton, 1991), with a somewhat greater emphasis on the low road.

Obviously IBM could have chosen the synoptic approach or the incremental approach exclusively, or placed greater emphasis on the synoptic approach than on the incremental approach. Exclusive or primary use of the synoptic approach may not have been appropriate or feasible at that time since the concept of a GIW was still in its infancy, and very little was known about it then. Today, on the other hand, with greater maturity and understanding of GIWs, it may be feasible and appropriate to emphasize the synoptic approach more than the incremental approach, if not exclusively the former.

Although incremental, the strategy was purposive. The process was kept under control through constant feedback and iteration between the teams and the customers. The organizational learning that occurred through these processes appears to have been of both the first and second order. First-order learning was focused on the design, development and implementation of the GIW itself. The second-order or meta-learning focused on the process of learning. For example, when the teams could not determine what charts and reports needed to be constructed from the data, they reversed the process and deconstructed the data required from the charts and reports that were desired by the users. Such flexibility and ability to change gears instead of doggedly pursuing a given approach is a sign of second-order learning. The political and financial support for the project and the organizational culture provided the safety net within which the two types of learning could occur with minimal fear and risk to the participants. They were necessary to the success.

Strategies for movement from data to information along the semiotic scale

The semiotic dimension of the development of a global information warehouse is evident in Fig. 2. Roughly, the bottom-most level in the figure with the databases represents the syntactic level of processing. The intermediate level, with the chart facility, the query support and the information delivery, represents the semantic level of processing. The topmost level, consisting of management, knowledge workers and the executives, represents the pragmatic level of processing. Thus upward movement in the figure partly represents information generation; downward movement partly represents information dissipation. (Partly — because the figure does not represent the morphological level and the transition between morphological and syntactic levels.) It must be noted that the overwhelming emphasis in the case is on information generation, not on information dissipation. However, both facets of the semiotic cycle played an important role in the design, development and implementation of the GIW as explained below. The semiotic strategy, if there was one, appears to have been purely incremental. The existing databases formed the basis for the data warehouse, and the data warehouse in turn formed the basis for the information warehouse. There is no evidence of top-down, needs-driven, synoptic information architecture.

'Language' appears to have been the primary barrier to movement along the semiotic scale. The intermediaries' translation of the end-users' requirements for the system analysts, and of the system analysts' design for the end-users, played an essential role in the successful movement along the semiotic scale. This, together with the constant communication, iteration and feedback within the cross-functional teams and between the teams and the users, appears to have played a key role in the success of the system.

However, the semiotic development took time. The intermediaries, somewhat like anthropologists exploring new cultures, had to gain the trust of a variety of groups and learn their 'languages'. In the long run, though, their 'multilingualism' appears to have paid off, in the form of both an accelerated learning curve and greater integration of needs across the users.

The development team collected all the necessary data elements at the lowest level of detail, and from that collection developed higher-level objects. These objects were then validated by the customers through many iterations. Thus the information objects were constructed from the existing data based on user needs. When this did not work, the data objects were deconstructed from the desired information objects. Thus there was a combination of forward and backward semiotic cycles in facilitating the movement along the semiotic scale.

The information objects were traditional objects such as charts, tables and reports — familiar to the users and the design, development and implementation teams. In a sense this would be the natural consequence of the incremental strategy. There was no effort to design new objects. This may have been due to many other reasons too. Partly it may have been due to the limitation of the technology at that time. The ability to create hybrid objects easily — such as graphs, tables and reports combined in one — has emerged only recently. The absence of focus on the design of new objects may also partly have been due to overheads in terms of education and training of the users that would have been involved in their implementation. In addition, it may have been partly due to the absence of a repertoire of standard information objects and therefore a need to create one, which would not leave much time or many resources to create innovative information objects.

Obviously today the situation is different. Repertoires of standard information objects are available for many industries. The technology has advanced too, and because of it there is the opportunity to create a greater variety of objects. Thus one would have greater flexibility and freedom in the design, development and implementation of information objects today. Not only that, but it would be possible to impose at least a quasi-synoptic perspective on the information objects.

Organization of objects

No explicit consideration appears to have been given to a strategy for the organization of objects. That is not to suggest that there was no organization.

From the overall architecture shown in Fig. 2 it would be reasonable to presume that these objects were organized functionally. They do emphasize mapping cross-functional relationships between data and information objects. The GIW was intended to move out of the functional organization, but built these relationships by starting with functional systems. The design and development process was, in part, set up to accomplish this.

The business segments and business processes were also used as axes to organize the objects, as shown in Fig. 3 and explained in the corresponding section earlier. This mapping, and the process for accomplishing it, was designed to ensure that designers' organization of objects corresponded with their customers' mental maps and needs.

A variety of data formats were used. The relational tables were multi-indexed to improve access and purposive searching with a process focus. Further, the users manipulated the relational tables, moving form process data to process information. Lastly the query and decision-support tools promoted the semipurposive exploration of data: the data information ad hoc transformation.

Thus an encyclopaedic strategy for organization of objects was adopted, although somewhat unconsciously. The classification was based on the business processes. In the future one could explore alternative methods of organization within an encyclopaedic strategy, and a theoretical framework to move the organization of objects to the level of a library.

Contextual factors

Many contextual factors played an important role in the successful implementation of the GIW; these have been described in 'Implementation challenges' above. They are the financial and political support for the project, the culture of the enterprise, the available skills, and user training. These factors are necessary but not sufficient for the successful implementation of any information systems project. They are not necessarily unique to the GIW implementation at IBM described in the case. In fact, these factors are recognized as important contextual elements likely to promote the successful implementation of information technology innovations in organizations. The fact that these factors surfaced in our case study and their description helps to put them in perspective for GIW implementation initiatives. All these factors can be significant facilitators of or barriers to expanding the system boundary, advancing the semiotic level of objects, and organizing the information objects in the development of a GIW. The case describes the particular ways in which IBM dealt with these issues. Although it is not the only way to deal with them, it is one way.

CONCLUSION

A global information warehouse is a step in the evolution of a global knowledge library. We present a framework for conceptualizing the evolution of systems such as these. The framework is based on three dimensions, namely the system boundary, the semiotic level of objects, and the organization of objects. Progression along the three dimensions can be a consequence of different strategies. These alternatives are discussed.

The case study describes the strategies used in the implementation of a GIW at IBM. It is subsequently analysed in light of the framework proposed in the paper. The evolution towards a global knowledge library is quite clearly a complex process. There are many alternative strategies, but there are as yet no criteria to choose a particular strategy. It is hoped that the framework proposed in this paper will help researchers and practitioners to conceptualize the general trajectory, if not the specific path, of evolution of such systems within an organization.

REFERENCES

Allen, B.R. & Boynton, A.C. (1991) Information architecture: in search of efficient flexibility. *MIS Quarterly*, **15**(4), 435–445.

Ambrosio, J. (1993) Tales from Warehouse. *Computerworld*, **27**(17), 55.

I/S Analyzer (1993) **31**(8), 1–16.

Inmon, W.H. & Osterfelt, S. (1992) Data patterns say the darnest things about your business. *Computerworld*, **26**(5), 73, 76–78.

Lucas, H.C. (1981) *Implementation: Key to Successful Information Systems*. Columbia University Press, New York.

Lucas, H.C., Ginzberg, M.J. & Schultz, R.L. (1990) *Information Systems Implementation: Testing a Structural Model*. Ablex Publishing Corporation, Norwood, NJ.

Orr, K. (1991) IBM hail for '90s: free jailed data. *Software Magazine*, **11**(9), 51–57.

Ramaprasad, A. & Rai, A. (1996) Envisioning management of information. *Omega: International Journal of Management Science*, **24**(2), 179–193.

Scott Morton, M.S. (ed.) (1991) *The Corporation of the 1990s*. Oxford University Press, New York.

Zmud, R.W. (1983) *Information Systems in Organizations*. Scott Foresman and Company, Glenview, IL.

Zultner, R.E. (1993) TQM for technical teams. *Communications of the ACM*, **36**, 78–91.

Biographies

Kathy A. Loeb is currently a Partner with Deloitte & Touche Consulting Group focused on Information Technology. She was formerly with the IBM Corporation (1982–97) and invested a significant portion of her career with IBM focused on the design, implementation and successful deployment of enterprise-wide data warehouses linked to executive, decision-support systems. Kathy, also a certified Project Management Professional (PMP), specializes in the healthcare industry and in delivering client-driven professional information technology consulting services to healthcare providers and payers. These services include strategic information systems planning, electronic commerce application design and development, infrastructure assessment, planning and deployment, intranet/extranet/ internet planning and enablement, data warehousing/ information access and PeopleSoft package implementations.

Arun Rai is Associate Professor in the Department of Decision Sciences at Georgia State University. He received his PhD from Kent State in 1990. His present research interests include the diffusion, infusion and impacts of information technology, information technology design for information and knowledge management, and management of unstructured processes such as innovation, product development, decision making and systems development. Dr Rai has published several articles on these and related subjects in journals such as *Accounting, Management and Information Technologies*, *Communications of the ACM*, *Decision Sciences*, *Decision Support Systems*, *European Journal of Information Systems*, *Journal of Management Information Systems*, *Omega* and several others. Major corporations, such as Bozell Worldwide, Chrysler, IBM and Comdisco, have funded some of his research work. He is an associate editor for *MIS Quarterly* and *Information Resources Management Journal* and a department editor for the *DATA BASE for Advances in Information Systems*.

Arkalgud Ramaprasad is a Professor in the Department of Management and Director of the Pontikes Center for Management of Information. He teaches and conducts research in strategic management and management information systems, and promotes industry–university cooperation in education and research related to management of information. He obtained his PhD from the University of Pittsburgh in 1980. Prior to that he obtained his MBA from Indian Institute of Management, Ahmedabad, India, and BE (Electrical) from the University of Mysore. He has published in *Behavioral Science*, *Management Science*, *Academy of Management Review*, *Omega*, *Decision Sciences* and other journals. He has received research grants from major corporations such as Comdisco, Chrysler, IBM and Bozell Worldwide.

Srinarayan Sharma is an Assistant Professor of Management Information Systems at Oakland University at Rochester. He received his DBA from Southern Illinois University at Carbondale. His research interests are in the diffusion of information technology, knowledge management, and data warehousing. His previous research has been published in or is forthcoming in the *Communications of the ACM* and *International Journal of Computer Applications in Technology*.