Integrated Business Information Systems Development: A Management Issue Not A Technological Solution

Ralph R. Bravoco, Information Systems, Texas Tech University

Abstract

An overview of the role of enterprise issues in Business Information Systems Development is presented. The importance of information flow and data standards in the development of an enterprise view is discussed and the need for management control and coordination of information integration is emphasized. In addition, a tested phased approach to information architecture development and the components of this information architecture are presented.

Introduction

The success of information system integration is directly dependent upon the extent to which intra-firm systems can be logically consolidated. This consolidation requires an understanding of information flows and the establishment of data standards. These, in turn, provide the critical basis for the development of a management approach to information system integration.

The purpose of this paper is to provide an overview of how to manage information systems integration developed under the U.S. Air Force's \$150 million ICAM (Integrated Computer Aided Manufacturing) program. The same methods were then successfully applied to the Army's ECAM (Electronics Computer Aided Manufacturing) Program, the Navy's Ship Technology Program, and all Air Force Technology Modernization Programs. Currently, major corporations world wide are incorporating the same procedures.

Information Flow Within An Enterprise

There are three widely different viewpoints of the information flow within a Business enterprise [2]. For Business Information Systems (BIS) to

be unsuccessfully used within the enterprise these views must be understood and coordinated.

The first view defines the demand for information. It is the user's view of BIS, determined by the enterprise's market environment and its various product and business life cycles. These create a need to know and a need to control by the user community, in turn creating the demand for information and subsequently, for enterprise-wide industrial and computer automation.

The second view considers the supply of information, and can be called the technology view of BIS. This view is created by pressures on the providers of technology. It contains all of the computer hardware, system software, turnkey applications, communications facilities, and databases used in a BIS environment.

The third view considers the integration and control of information and can be called the enterprise view of BIS [1]. This view provides a control structure that can maintain alignment between the dynamic user and technology views, while at the same time providing for the integration and

consistency required by the enterprise as a whole.

In the past, the first two views have been closely interwoven, to the point that it has almost become impossible to distinguish between them. What, for example, is MRP (Material Requirements Planning)? Is it a user concept or a technology concept? What is CAD (Computer Aided Design) or CAM (Computer Aided Manufacturing)?

Because of the haste to automate, user views have in fact been driven by technology views of BIS. In business's classic pragmatic style, automation has occurred from the bottom up. The result is that specific user views have been fitted to specific technology views, and--as the vendors had planned--are now hostage to them.

This fragmented approach to automation has produced islands of automation. Typical islands include CAD, CAM, MRP, group technology, shop floor control, cost, process planning, bill of material, purchasing, and so forth. A typical enterprise may have automated up to 50 independent functional areas, each with its own hardware, system software, communications, files, databases, and so on. This is the situation that Integrated Business Information Systems (IBIS) aims to remedy.

Since the word integration is so critical here, we should probably define it. Two (or more) things that are integrated have common parts. These common parts are leveraged to provide economies and benefits. So where are the common parts in IBIS, and to whom do they belong? Are they technology parts that belong to the vendors? Are they user parts? The answer is that common parts belong to the enterprise, and give rise to the enterprise view of IBIS.

The enterprise view of IBIS contains planning and project management

procedures, system and data standards, budgeting and performance controls, and organizational responsibilities. IBIS is not possible without an enterprise view that defines what will be shared, why, and how. Through its standards and procedures it answers questions like, will data be shared, and if so how? Will machinery be shared? Programmers? Communications? Reports? If integration is the major issue of IBIS, it must come from somewhere. Hardware and/or software vendors would like integration to be primarily a technology issue. But, since most businesses are already heterogeneous technical environments, it is ridiculous to assume that any single vendor will provide blanket technological integration. Even if one could, nobody could use it because nobody has the luxury of automating from scratch. There is no technological panacea for the problems of multiple, isolated systems.

Integration is a management issue, not a technical issue. It must be established through methods and standards. Integration is not a magic wire running among various machines. It is a managed infrastructure of what runs on the machines, regardless of their manufacturer.

For IBIS to become a reality, it must become a management style. Integration occurs because automation projects are executed from a purposeful plan, by organized, trained people using consistent tools and techniques, working within specific business standards. It cannot be bought.

Today's enterprise views of IBIS are, with few exceptions, anemic at best. This is why we have so many islands of automation. The typical automation strategy in business is "everyone out for a pass!" Managers are lured by instant gratification from technological fixes, and they are coerced into patches by financial justification procedures. They have little incentive to put forth energy to plan and

control. The idea of spending \$500,000 to \$1 million putting plans, project management systems, data standards, special software development tools and procedures in place to manage IBIS is still odd to many. They ask, "What about our centralized dp shop? Isn't that enough?"

It's not, of course, but it is a place to start--along with the CAE/CAD/CAM department. An enterprise view of IBIS must first integrate the people who are doing IBIS things. If it doesn't start there, it will surely fail.

The path to a IBIS enterprise view is to set up a IBIS program. Such a program is like a zero based budget program. Its objective is to change management style by establishing a framework within which industrial automation projects are defined, funded, managed, and coordinated. This framework requires specific mechanisms for planning, financial control, project selection and justification, project management, and project performance monitoring.

The role of the enterprise view of IBIS is to ensure the appropriate levels and types of integration. The most important concept in integration involves the use of standards. There are two types of standards: technical and data.

Technical standards are set by and for the whole enterprise. They define what is sometimes called the computer system's architecture. At a minimum, standards for technical procurements are needed in the following areas: tele-communications (e.g., IEEE 802.4), database management (e.g., SEQUEL 2), and graphics exchange (e.g., IGES 2.0).

Unlike technical standards, data standards cannot be obtained from a standards committee. They must be defined and maintained by the enterprise itself. An enterprise view of IBIS that includes data standards is said to

be data driven because it uses those standards to control software package procurement and in-house database development.

Data Standards

Data standards are used to manage shared data, i.e., data needed by many people. The data most often shared in businesses are product and process definition data, which is why product definition databases are the fastest growing segment of IBIS. Business planning, cost, and control data are also highly shared and must eventually be brought under the control of data standards. There are two types of data standards: business rules and data element standards [2]. These are defined from data architectures constructed using an information modeling methodology such as the Air Force's IDEF44155 C[5C], Carlson's BIAIT C[8C], IBM's BSP C[10], Bachman's DSD C[3C], and Chen's E-R Model C[9C]. The selection of the methodology is a crucial enterprise-level decision. The resultant standards are used to control the most critical data in the enterprise. Section 4.0 of this paper presents the basic concepts of any of these Information Architecture technologies to help you select one for your IBIS development.

Data standards are mandatory to achieve the levels of data integrity and consistency needed to run custom shops efficiently and effectively. The target for data integrity in business is .999, not just for the operational control systems, but for the management and strategic levels as well. This is why a main responsibility of the enterprise view of IBIS is to establish and maintain control over highly shared engineering, manufacturing and administrative data. These data are deployed on the computers in the technology view and employed by users in the user view.

This control of engineering, manu-

facturing, and administrative data is usually known as the Data Base Management System (DBMS) issue. This issue requires knowledge of two things: a consistent set of database management system (DBMS) technologies across different vendor hardware, and consistent definitions of the data being stored and manipulated by the machines. These are not the same problem, and they are only loosely related.

The DBMS issue is clearly being driven toward relational technology. Most vendors (including DEC, IBM, HP, and CDC) are developing relational database management systems with the functions necessary to optimize data independence, redundancy, integrity, accessibility, security, shareability, and performance in a distributed homogeneous (same hardware vendor) environment. Other vendors, like General Electric, Martin Marietta, and Computer Corp. of America, are attempting to develop powerful data dictionary systems to provide the same capabilities for managing data in a distributed heterogeneous environment.

The data definition issue is generally being addressed from the enterprise view, but it's important to note one thing here. The typical business has at least five (and up to 35) different types of part numbers, three (and up to 10) types of bills of material, and from six to 20 types of costs. It will have seven different types of changes and 16 different types of schedules. The reason is that it develops (or buys) application packages without any consistent strategy for standardizing these simple concepts. The outcomes range from minor confusion to chaos that threatens the ability to produce products and manage assets. Why? Because of inconsistent and inaccurate data.

Objectives of an Information Architecture

Anyone working with information,

such as a programmer or data base manager, is helped by a clear understanding of the structure of the information to be processed. For small tasks the subject may be easily perceived. For larger tasks, however, a method of recording the structure is desired. Just as the civil engineer needs maps and three dimensional models to plan roads which are scenic, safe and economic, the data processor needs a map of the information "world" to guide his efforts. An Information Architecture provides that map. It is important that the format of the "map" permit reviews by users who have no prior training in data processing. Because users can validate the structure of the information, misunderstandings between users and data processors are reduced. An Information Architecture provides a solid base for the contract, or handshake, between the users of the information and those who will store and process it.

Information Architecture Basic Concepts

An Information Architecture should be developed using a comprehensive methodology for describing and analyzing the information of an object system C[5,6,7,11C]. By comprehensive methodology is meant a coherent, integrated set of methods and rules that constitute a disciplined approach to analysis and design, built upon a foundation of closely interrelated concepts.

Fundamental Concepts of an Information Architecture's Methodology

Six fundamental concepts of an Information Architecture's methodology are:

1. Should attack a problem by building a model or a representation of the problem, which expresses an in-depth understanding of an enterprise such as a business, school or governmental unit.
2. Would use a limited set of components to build models.

3. Should differentiate as much as practicable between the inherent structure of the information and the storage media (e.g., forms, reports, tapes, discs) on which the information appears.

4. The language of the methodology should be a diagramming technique which would show component parts, interrelationships between them, and how they fit into a structure.

5. The methods should support disciplined, coordinated teamwork, which would be required in order to produce results which reflect the best thinking of an enterprise.

6. The methods should require that all analysis decisions and comments thereon be in written form.

Concept 1: Understanding via Model Building

The methodology should be applied to a problem by building on paper a model, or Information Architecture which would express an in-depth understanding of what the problem is. The methodology should provide the unifying, consistent record of the way in which the enterprise would view all its information.

A single Information Architecture could guide the development and integration of many information systems within an enterprise. This would be possible because each of the systems would operate on a subset of the full Information Architecture, and because the subsets should overlap.

Concept 2: A Limited Set of Components

The methodology should have only three primary elements: classes of things (or entities) to which information relates; classes of kinds of information (or attributes) about those things; and classes of relations between those things. Classes of entities might

be: the class of employees, the class of machines, or the class of production orders. Classes of attributes might be: employee date of birth, machine serial number, or product on order quantity. Classes of relations might be: assignment of employees to departments, or authorizations to make specified part numbers by specified production orders.

Concept 3: Differentiation Between Structure and the Storage Media

It is not uncommon for information requirements to be portrayed in terms of report formats or data base layouts. Such portrayal is useful, but subject to change and duplication. Behind all of the reports stands a single structure by which a vendor, his address, and the parts he is authorized to deliver are related. An Information Architecture should document that structure.

The structure captured by the Information Architecture should not vary with the design of a new form or the rearrangement of a data base. It should be the relative permanence of the Information Architecture that would make it applicable across systems, forms and data base layouts.

Concept 4: Graphic Format of Model Representation

The Information Architecture should consist of some number of entity classes connected by lines and symbols to represent the relationship between entities being represented. The combination of lines and symbols should represent the basic relation class syntax to be employed. The graphic format should portray the entity classes and the relation classes through which they would be linked. The graphic portrayal should then be supplemented by listings of attribute class information. Most relation classes should be shown as relating to more than one other entity class. For each entity class, one or more of its attributes which uniquely identify an entity, sho-

uld be selected as a key class. Such key classes should be listed within the graphic entity class symbol. Graphic presentations should be particularly effective in giving users a visual image of the structure they would be asked to validate.

Concept 5: Support of Disciplined Teamwork

Analysis of complex systems requires the coordination of the creation, modification, and verification of the specification of information content. No single person can comprehend completely every aspect of a complex system within the time limits usually imposed. Even if this were possible, it would require an undesirable dependence upon one person. Analysis and design requires disciplined, coordinated teamwork. Consequently, the insights and view points of project personnel must be communicated effectively at every step and level of analysis to insure the Information Architecture produced reflect the best thinking of the team. Adequacy and quality must be assured by regular, critical review, thus creating evolutionary changes and corrections as necessary.

Continuous and effective communication requires uniform, understandable, complete, and current documentation throughout a project. This documentation must be produced, as the analysis or design decisions can be seen in context and challenged while alternative approaches are still viable. The documentation provides the reasons for decisions and vastly improves the visibility of the project throughout the team and to management. The evolving Information Architecture with its supporting text would be such a set of documentation.

Throughout a project the draft versions of the Information Architecture produced should be distributed to one or more other project members (usually other system developers) for

review and comment. The discipline of the Information Architecture's methodology should require that each person expected to make comments about a particular Information Architecture diagram would make them in writing on the diagram and should submit them to the originator of the diagram. When these commenters would finally reach agreement with the originator, the updated diagram should next be reviewed by a Technical Committee of senior analysts. Such an approval cycle would continue upward in the organizational structure until the diagrams and eventually, the entire Information Architecture, would be officially accepted. During the process, incorrect or unacceptable analysis and design results would usually be spotted early, and oversights and errors would be detected before they can cause major disruptions.

The documentation produced, as the Information Architecture evolves, would provide continuing visibility into the status of the project. Technical management should be able to understand the higher levels of the Information Architecture so as to be able to study the system top-down to whatever level of detail is relevant.

Concept 6: All Decisions and Comments in Written Form

The Information Architecture's methodology should include procedures which would retain written records of all decisions and alternate approaches as they unfold during the project. Commenters should be able to document their suggestions directly onto copies of the Information Architecture. System developers should be able to respond to each comment in writing on the same copy. Suggestions should be accepted or rejected in writing along with the reasoning used. As changes and corrections would be made, outdated versions of diagrams should be retained in the project files --nothing should be thrown away.

This documentation process of analysis would record why particular decisions were made and what factors influenced them. Possible future extensions of the system as well as system maintenance would be able to make use of previously recorded analyses and design decisions. The process of creating this form of Information Architecture and retaining it in an official System File as the system would be developed should avoid a documentation phase after the project would be completed. Since the actual system development would be carried out using the Information Architecture methodology, the documentation would precisely match the final, working system.

A Phased Approach to Information Architecture Development and Use

As described above a practical way to approach the problem of integration of a systems information would be to develop an Information Architecture prior to designing and building the corresponding information system. However, it would only be the first step. A more complete approach should be:

1. build an integrated Information

- Architecture;
2. design a data base(s) from the Information Architecture;
3. implement and install the data base(s) and associated functional and procedural components.

Summary and Conclusions

The key to IBIS is integration. The success of IBIS within a firm is directly dependent upon the extent to which intra-firm systems can be integrated. This information integration is a logic problem, requiring a recognition of information flows, establishing data standards, and developing an information architecture. In addition to discussing the critical importance of each of these factors to IBIS success, several information modeling methodologies are identified and their conceptual steps summarized. Six fundamental concepts are identified as providing the foundation for an information architecture methodology. Each of these steps is discussed and its importance to the successful development of an information architecture described. Finally, a phased approach to information architecture development and use is suggested.

References

- [1] D.S. Appleton, "The State of CIM," *Datamation* (December 15, 1984).
- [2] D.S. Appleton, "Business Rules, the Missing Link," *Datamation* (October 15, 1984).
- [3] C.W. Bachman, "Data Structure Diagrams", Database 1 (Summer, 1969).
- [4] F. Boraz, et al., "The Role of the Computer Based Information System in ICAM," Proceedings of the Sixth Annual Integrated Computer Aided Manufacturing Industry Days (January 17-20, 1982).
- [5] R.R. Bravoco and S.B. Yadav, "A Methodology to Model the Information Structure of an Organization," *The Journal of Systems and Software* (January, 1985).
- [6] R.R. Bravoco, "Information Systems Engineering--An Overview," Society of Manufacturing Engineer CIMCOM Conference Proceedings (March 13-15, 1984) and SME Technical Paper MS84-183.