

1993

Incorporating object analysis and design into a structured information systems development life cycle

David C. Wallace
Illinois State University

Follow this and additional works at: <http://scholarworks.lib.csusb.edu/jiim>

 Part of the [Management Information Systems Commons](#)

Recommended Citation

Wallace, David C. (1993) "Incorporating object analysis and design into a structured information systems development life cycle," *Journal of International Information Management*: Vol. 2: Iss. 2, Article 4.
Available at: <http://scholarworks.lib.csusb.edu/jiim/vol2/iss2/4>

This Article is brought to you for free and open access by CSUSB ScholarWorks. It has been accepted for inclusion in Journal of International Information Management by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

Incorporating object analysis and design into a structured information systems development life cycle

David C. Wallace
Illinois State University

ABSTRACT

An object approach to information system development should be incorporated into the structured system design life cycle (SDLC) during the analysis and design phases. The tools of the structured approach (data flow diagrams, data structures, data stores, etc.) help to identify the relationships that exist between the data stores and the entities (objects) within an information system. This is only a starting point at which the IS developer further evaluates the informational needs that are required within the information system. A listing of the data structures is generated to associate the specific data elements that comprise each data structure. A series of logical steps (normalization) can be used to remove the redundancy and inflexibility within each data structure. The end result of the normalization process is the generation of an entity-relationship diagram (E-R diagram). The E-R diagram can be used as a reference point in which to apply a series of object oriented analysis and design steps to insure that information collected on objects throughout the organization is properly integrated into the SDLC. A revised diagram is subsequently used to modify the data flow diagrams which will be the basis for the design of the new system. Thus, the object approach to information system development is an integral component of the structured system design life cycle.

INTRODUCTION

Information systems (IS) development has gone through several changes over the past twenty years. These changes were part of an evolutionary process involving the systematic approach to information systems development—traditionally referred to as the System Design Life Cycle (SDLC). Generally, these changes can be grouped into three stages: (1) structured methodology, (2) Computer-Aided Software Engineering (CASE), and (3) object orientation. Each stage helped to define an approach to information system/software development by placing more emphasis on analyzing and designing the solution to users' problems and less time on generating computer code and revising software that did not meet the needs of the users. The structured stage helped to identify a methodology that focused on the goals and objectives of the organization and the underlying information system and subsystems that assimilate them into their related activities. With the introduction of Computer Aided Software Engineering (CASE) tools, the IS developer was able to automate the SDLC and the structured methodology. Currently, the object stage has emerged as an opposing methodology to the structured approach to information system development. Yet, by incorporating an object orientation into the structured approach to the SDLC, the IS developer can achieve the benefits of both structured methodology and object orientation.

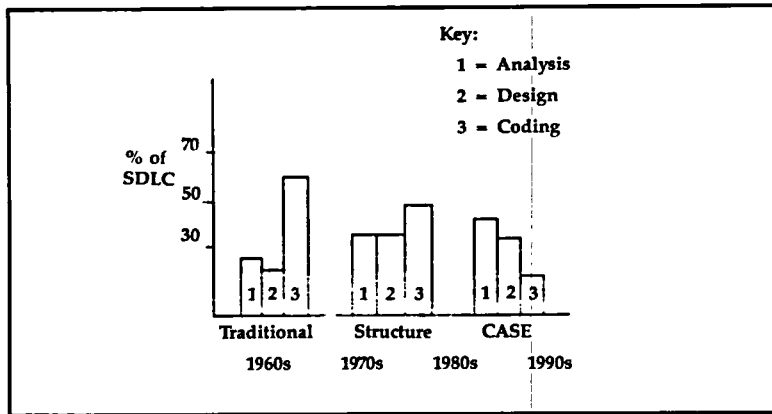
STRUCTURED METHODOLOGY

During the 1960s and early 1970s the development of information systems was an "undisciplined" approach resulting in poorly designed and implemented systems that did not meet the needs of the users who had to interact with them. These systems were often very difficult to control and to maintain. Information systems and software development was a costly effort in which supply fell short of the heavy demand for IS services. The IS developer (often the senior computer programmer) spent relatively little time on identifying users' needs and requirements, and more time on generating computer code (Souza, 1991). There was tremendous pressure to reduce time and cost by rushing the development process in order to place a software product into the production cycle of the organization. This resulted in more time and money spent on testing and maintaining software that did not fully meet the needs of the user who requested it (Kendall, 1992).

The structured stage established a more disciplined approach to the IS development process. This approach focused on the identification of the goals and objectives of the organization and the underlying information system and subsystems that assimilate them into their related activities. IS developers incorporated the structure approach into a series of steps or phases that would become a blueprint for the development of future information systems. This blueprint is generally called the System Design Life Cycle (SDLC). The various phases of the SDLC are: (1) problem definition, (2) analysis, (3) design, (4) implementation, (5) installation, and (6) review (Powers, 1990). The structured approach uses a modeling tool called the data flow diagram to reveal the activities and data flows within an information system. This model becomes the basis for both understanding and recommending improvements to the existing system. System flowcharts and structure charts are generated from the data flow diagrams to provide more specific details needed to generate the computer code (if customized coding is required) during the implementation phase of the SDLC.

The initial three phases (problem definition, analysis, and design) are generally considered the front-end phases where the needs and requirements of the users are evaluated with respect to the goals and objectives of the organization and the information system under investigation. By focusing on the front-end phases of the SDLC, the IS developer is able to better meet the needs of the user and significantly reduce the costs associated with testing and maintaining information systems (Kendall, 1992). The structured stage helped to change the SDLC so that more time was spent on the analysis and design phases and less on the coding activity. Figure 1 demonstrates this change.

With the introduction of Computer Aided Software Engineering (CASE) tools, the IS developer was able to automate the SDLC which allowed him/her to devote even more time to the analysis and design phases (see Figure 1). CASE technology has produced a wide range of automated tools which provided the IS developer many advantages: (1) emphasis on the analysis and design phases, (2) automated documentation, (3) automated auditing and design capabilities, (4) computer code generation, (5) reverse engineering, (6) design and code reusability, (7) prototyping, and (8) project administration.

Figure 1. Comparison Software Development (Souza, 1991)

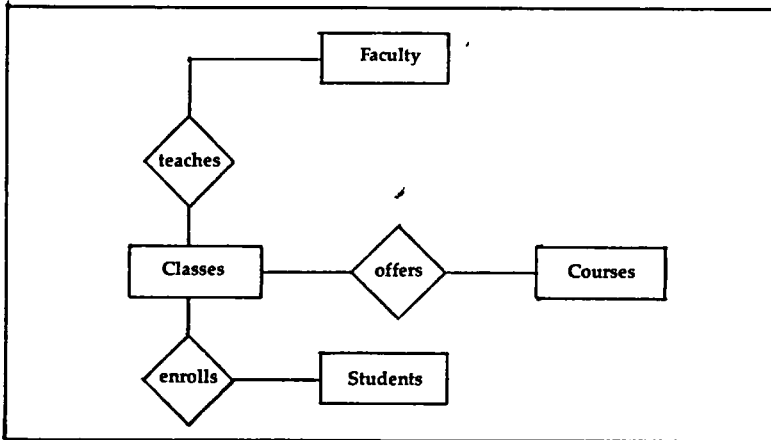
CASE is not only a productivity tool with which to help IS developers build better information systems. CASE tools can help top executives build and adjust models of their organization in order to meet strategically the demands of a rapidly expanding and competitive international market (Gibson, Snyder, & Carr, 1991). The power of the CASE tool lies in its ability to model information flows throughout an organization. This modeling enables managers at all levels within the organization to build and adjust the various information systems that comprise the organization. They can utilize the CASE tool as a prototyping mechanism to communicate and coordinate strategic initiatives more efficiently. This capability allows an organization the ability to seize the initiative and quickly enter a particular market with a new idea. The ability to adjust product lines or services and quickly compete in a market will give the organization a necessary strategic advantage with which to survive (Zimmerman, 1990).

OBJECT STAGE

The object stage evolves from the movement to identify information an organization collects on things or objects (called entities) that it considers important—for example, customers, suppliers, employees, inventory, and sales (Gane, 1990). Figure 2 illustrates a brief example of an entity relationship diagram for a student registration information system. In this example the student registration system collects information on faculty, courses, classes, and students. The relationship between these entities or objects is illustrated by a diamond. Thus, when a student schedule is produced within this information system, keys to each of the entities are used to access the information concerning that entity. For example, the key to the student entity might be his/her social security number, and this number is used to access

his/her address, phone number, and major. Furthermore, the key to the class file might access class section, class meeting time, and class meeting place. By using both keys, a schedule can be generated listing the student and enrollment information.

Figure 2. Sample Entity-Relationship (E-R) Diagram



Ideally, an organization can generate a logical diagram of its entities and the relationships that exist between them for all of its information systems. As a result, the organization would be able to manage effectively the information collected and used throughout all of its information systems. One of the major problems that organizations have is identifying and controlling the information needed to better achieve its goals and objectives.

INCORPORATING OBJECT INTO STRUCTURE

The object approach to information system development is not an opposing movement to the structure approach. By incorporating the object movement into the structure process, the IS developer can achieve the benefits of both methods. The SDLC provides the mechanism in which to place strategically both the structured and the object approaches. The IS developer would still use the data flow diagram to model the existing system. The processes on the data flow diagram help to identify not only the activities that comprise the information system under investigation, but also the relationships that exist between the data structures. The evaluation of these relationships can, along with other considerations, form the basis for new improvements on the existing system.

A common data structure is a data store which holds information about things that are important to the information system (very similar to entities on the E-R Diagrams). In the process of analyzing the data flow diagram and making recommendations on a new system, the IS developer should generate an E-R model of the information system as a tool to evaluate the data stores on the data flow diagram. In order to generate an effective E-R model, the

IS developer should determine which data stores are necessary in order to achieve the objectives of the information system. This can be accomplished through communications with both management personnel and users within the information system. Once the data stores are identified, the specific attributes or data elements of the data stores can be determined and listed in a data structure format. Figure 3 illustrates an example of several data structures used in a car rental business. During the analysis phase of SDLC, the IS developer would identify the data stores along with the data structure contents for the information system under investigation. He/she would also identify any relevant data stores used by other information systems that might impact the current information system. Figure 3 identifies two possible data stores that might have an impact on the current system: (1) CAR_FILE, and (2) EMPLOYEE_FILE.

Figure 3. Example Data Structures for Car Rental Business

Relevant Data Stores From Other Information Systems:

CAR_FILE = (ALL CARS IN BUSINESS)

EMPLOYEE_FILE = (ALL EMPLOYEES IN BUSINESS)

The Car Rental Information System:

CUSTOMER_RENTAL_FILE = CUST# + CUST_ADDR + CUST_PHONE +
(CAR# + CAR_FEE + CAR_MAKE + CAR_MODEL + CAR_COLOR)

EMPLOYEE_CAR_FILE = CAR# + EMPLOYEE# + EMPLOYEE_NAME +
EMPLOYEE_JOB + CAR_MILEAGE + CAR_YEAR +
NUMBER_TIMES_CAR_RENTED + AVERAGE_NUMBER_MILES_RENT

Once the data structures for these data stores are identified, a series of steps can be initiated to build an E-R model or Data Model for the information system. These steps are often called the normalization process. Since the normalization process is long and complicated, this example has been limited to four steps in order to illustrate the relationship between structured analysis/design and object analysis/design. The first step in the normalization process is to eliminate any repeating groups of data elements within a given data structure by creating a separate data structure for the repeating group. This can be accomplished by concatenating the key of the original structure and the key of the new structure (Powers, 1990). Figure 4 illustrates this relationship.

The second step in the normalization process is to examine any data structure with concatenated keys. The data elements of these structures should be evaluated in order to determine their full functional dependency on both keys. In other words, the concatenated keys imply a certain purpose which the data elements must support and not other subordinate purposes. In the example, CUST# and CAR# imply the purpose of identifying a specific car rented by a specific customer. Yet, the attributes CAR_FEE, CAR_MAKE, CAR_MODEL, and CAR_COLOR serve a subordinate purpose of identifying the characteristics of a specific car. In order to accomplish both purposes, two structures should be created. Figure 5 demonstrates this process (Powers, 1990).

Figure 4. First Step in the Normalization Process

$$\text{CUSTOMER_RENTAL_FILE} = \text{CUST\#} + \text{CUST_ADDR} + \text{CUST_PHONE} + (\text{CAR\#} + \text{CAR_FEE} + \text{CAR_MAKE} + \text{CAR_MODEL} + \text{CAR_COLOR})$$

Repeating Group

Results:

$$\begin{aligned} \text{CUSTOMER_FILE} &= \text{CUST\#} + \text{CUST_ADDR} + \text{CUST_PHONE} \\ \text{CUSTOMER_RENTAL_FILE} &= \text{CUST\#} + \text{CAR\#} + \text{CAR_FEE} + \text{CAR_MAKE} + \\ &\quad \text{CAR_MODEL} + \text{CAR_COLOR} \end{aligned}$$

Figure 5. Second Step in the Normalization Process

$$\text{CUSTOMER_RENTAL_FILE} = \text{CUST\#} + \text{CAR\#} + \text{CAR_FEE} + \text{CAR_MAKE} + \text{CAR_MODEL} + \text{CAR_COLOR}$$

Results:

$$\begin{aligned} \text{CUSTOMER_RENTAL_FILE} &= \text{CUST\#} + \text{CAR\#} \\ \text{CAR_FILE} &= \text{CAR\#} + \text{CARE_MAKE} + \text{CAR_MODEL} + \text{CAR_COLOR} \end{aligned}$$

The third normalization step involves two evaluations. The first evaluation determines if any of the data elements in the data structures can be derived by other data elements in the same structure. This evaluation insures that the nonkey data elements are mutually independent of each other. In the EMPLOYEE_CAR_FILE data structure, the average number of rented miles can be calculated by dividing the number of times the car is rented by the car's mileage. The number of rented miles and the car's mileage are both data elements within the data structure. Thus, the AVERAGE_NUMBER_OF_MILES_RENT would be deleted from the data structure.

The second evaluation determines the full functional dependency of every data element on the key of its structure. Again, the object is to insure that each nonkey data element represents the purpose implied by the key of that structure. In the EMPLOYEE_CAR_FILE data structure, the key CAR# implies that the data structure identifies a specific car. After examining the nonkey data elements, several purposes can be identified: (1) identification of a specific car, (2) identification of the employee who is assigned to maintain the car, and (3) identification of a specific employee. Each purpose can be represented by a separate data structure. Figure 6 illustrates this process.

Figure 6. Step 3 in the Normalization Process

$$\begin{aligned} \text{EMPLOYEE_FILE_FILE} = & \text{CAR\#} + \text{EMPLOYEE\#} + \text{EMPLOYEE_NAME} + \\ & \text{EMPLOYEE_JOB} + \text{CAR_MILEAGE} + \text{CAR_YEAR} + \\ & \text{NUMBER_TIMES_CAR_RENTED} + \text{AVERAGE_NUMBER_MILES_RENT} \end{aligned}$$

Results:

- 1) $\text{CAR_FILE} = \text{CAR\#} + \text{CAR_YEAR} + \text{CAR_MILEAGE} + \text{NUMBER_OF_TIMES_CAR_RENTED}$
- 2) $\text{EMPLOYEE_CAR_FILE} = \text{CAR\#} + \text{EMPLOYEE\#}$
- 3) $\text{EMPLOYEE_FILE} = \text{EMPLOYEE\#} + \text{EMPLOYEE_NAME} + \text{EMPLOYEE_JOB}$

The normalization process produces additional data structures. Figure 7 summarizes the data structures that exist at this stage in the normalization process. After examining these data stores the IS developer should combine the data structures that are similar. The original CAR_FILE structure (from relevant Data Stores of other information systems) has the same data elements as the two CAR_FILE structures created during the normalization process. Thus, these data stores will be combined into one. Similarly, the EMPLOYEE_FILE of the original structure has the same data elements as the EMPLOYEE_FILE created in the third normalization step. This data structure should be combined with the original data store.

Figure 7. Normalized Data Stores - Third Step

Relevant Data Stores From Other Information Systems:

CAR_FILE = (ALL CARS IN BUSINESS)

EMPLOYEE_FILE = (ALL EMPLOYEES IN BUSINESS)

Normalized Data Stores (end of third step):

CUSTOMER_FILE = CUST# + CUST_ADDR + CUST_PHONE

CUSTOMER_RENTAL_FILE = CUST# + CAR#

CAR_FILE = CAR# + CAR_MAKE + CAR_MODEL + CAR_COLOR

CAR_FILE = CAR# + CAR_YEAR + CAR_MILEAGE +
NUMBER_OF_TIMES_CAR_RENTED

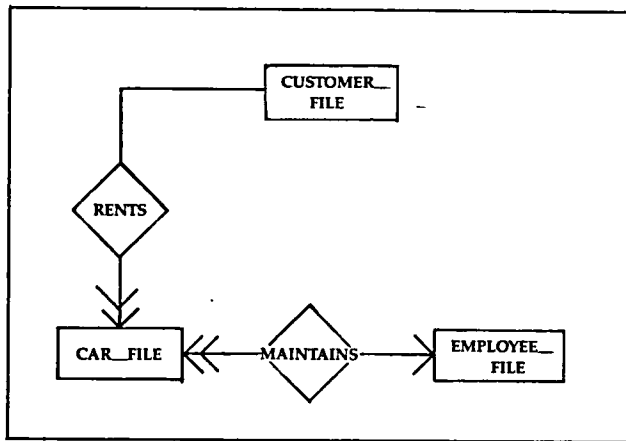
EMPLOYEE_CAR_FILE = CAR# + EMPLOYEE#

EMPLOYEE_FILE = EMPLOYEE# + EMPLOYEE_NAME + EMPLOYEE_JOB

The process of combining similar data structures is the fourth normalization step. The normalization process generated a data model for this information system which includes three basic files: (1) EMPLOYEE_FILE, (2) CAR_FILE, and (3) CUSTOMER_FILE, as well as two correlational files: (1) CUSTOMER_RENTAL_FILE, and (2) EMPLOYEE_CAR_FILE, (Powers, 1990). An Entity-Relationship Diagram can be generated to represent relationship

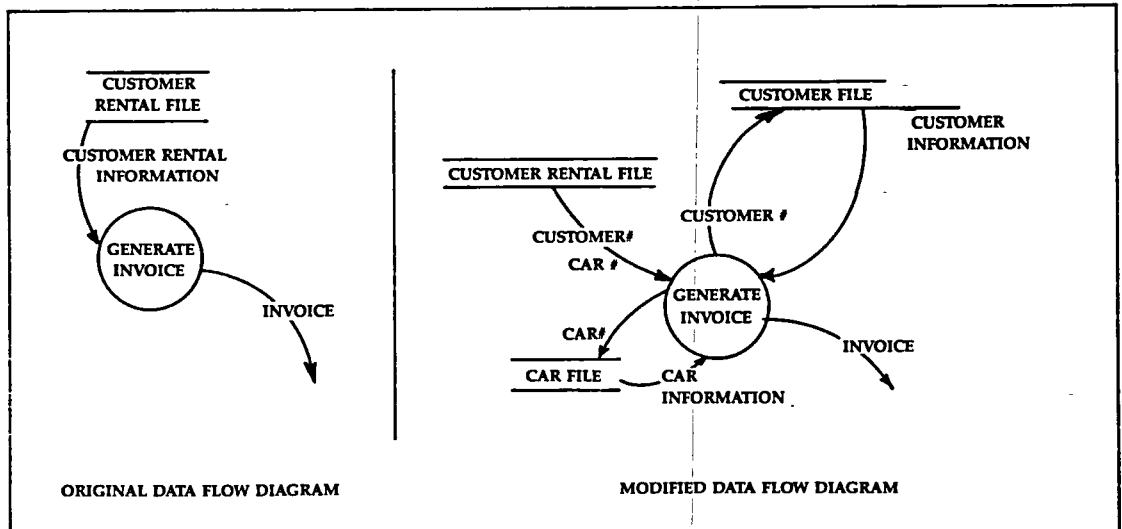
between the basic files which will be identified as entities and the correlational file which will be identified as relationships between the entities. Figure 8 represents the Entity-Relationship Diagram that can be generated from this example

Figure 8. Car Rental Entity-Relationship Diagram



This example was not intended to be a complete exhaustive study of the normalization process or the process of evaluating entity-relationship diagrams. The normalization process can involve many more steps especially when the data stores are numerous and complicated. The evaluation of entity-relationship diagrams can include additional activities that require review and feedback from both management and the database administrator. The output of this process should be the generation of an Entity-Relationship Diagram that allows the organization to control and maintain information in order to effectively achieve its goals and objectives. The IS developer can use the diagram to modify the existing logical data flow diagram generated during the analysis phase. Thus, the original data stores on the data flow diagram (Figure 3): (1) CUSTOMER_RENTAL_FILE, and (2) EMPLOYEE_CAR_FILE would change into correlational files with concatenated keys. A new file would be created called the CUSTOMER_FILE. Figure 9 illustrates the type of changes that would have to be made on the data flow diagram.

Figure 9. Modified Data Flow Diagram



SUMMARY

The object approach to information system development should be incorporated into the structured system design life cycle during the analysis and design phases. The tools of the structured approach (data flow diagrams, data structures, data stores, etc.) help to identify the relationships that exist between the data stores and the entities (objects) within an information system. This is only a starting point at which the IS developer further evaluates the informational needs that are required within the information system. A listing of the data structures is generated to associate the specific data elements that comprise each data structure. A series of logical steps (normalization) can be used to remove the redundancy and inflexibility within each data structure. The end result of the normalization process is the generation of an entity-relationship diagram. This diagram is subsequently used to modify the data flow diagram which will be the basis for the design of the new system. Thus, the object approach to information system development is an integral component of the structured system design life cycle.

REFERENCES

- Gane, C. (1990). *Computer-aided software engineering: The methodologies, the products, and the future*. Englewood Cliffs, NJ: Prentice-Hall.
- Gibson, M. L., Snyder, C.A., & Carr, H. H. (1991). Why CASE belongs in strategic business management. *Information Strategy: The Executive's Journal*, 7(2), 17-21.

- Kendall, P. A. (1992). *Introduction to systems analysis and design: A structured approach* (2nd Ed.). Dubuque, IA: Wm. C. Brown, Publishers.
- Powers, M. J., Cheney, P. H., & Crow, G. (1990). *Structured systems development: Analysis, design, and implementation* (2nd Ed.). Boston, MA: Boyd & Fraser Publishing Co.
- Souza, E. (1991, Winter). The impact of CASE on software development. *Journal of Information Systems Management*, 8(1), 18-24.
- Zimmerman, R. (1990, May 16). Information systems integration: A strategic asset for a corporation. *1990 Information Resources Management Association International Conference*, Hershey, PA (keynote speech).

Business visualization: More effective management tools for observation, dynamic measurement and control

Hal Records
Alan Olinsky
Bryant College

ABSTRACT

Increased global competition and continual pressure to produce more work with fewer resources has become the hallmark of the 1990s. Businesses are taking steps to meet these new challenges by downsizing for better control, using networks (Valovic, 1992) and data communications for rapid response, and purchasing personal tools such as word processing programs, spreadsheets, graphics and desktop publishing programs for an increase of manager productivity. But are these steps sufficient? In order to be more profitable in today's fast-paced and ever-changing environment, business managers at the strategic, tactical and operational levels must be able to visualize their operating data in a dynamic way. Business Visualization (BV) can be defined as the process of using computer based information systems to put business transactions and performance into a dynamic visible form that is readily understood by users (Marcus and van Dam, 1991). Our proposal is that business transactions and performance can and should be measured in a continuous rather than discrete manner.

THE CHANGING ENVIRONMENT

The process of business visualization is important because it affects the degree and speed with which managers respond to changes in the marketplace. Business happens at an accelerated pace in our world of global electronic communications. Stock markets in New York, London and Tokyo make the view of an eight-hour work day obsolete. Multinational firms operate in many time zones. Business is always happening somewhere and today's satellite telecommunicaitons, fax machines, modems, and computers make available a wealth of data and information that transcends traditional time and language barriers.

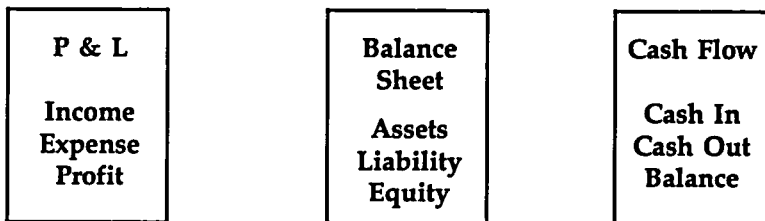
INFORMATION OVERLOAD

Since data can be collected, stored and moved quickly and inexpensively, businesses are collecting masses of data which result in information overload. The sheer volume of data coupled with a traditional dependence on paper reports has a crippling effect on the ability of managers at all levels to see the essence of business operations. Business is dynamic. It begs to be viewed in a non-static way.

TRADITIONAL VS. NEW VIEWS OF BUSINESS

Traditional accounting reports such as statements of cash flow, profit and loss statements, and balance sheets (as shown in Exhibit 1) are static. They portray business at fixed points in time, annually, quarterly, monthly, weekly and sometimes daily. Although these reports have use and value, are they sufficient?

Exhibit 1. Traditional Business Visualization

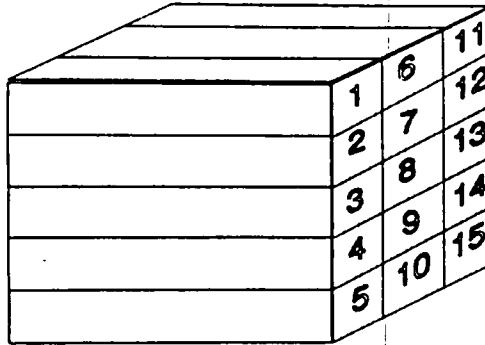


Do they make it easy for managers to see the living, breathing, changing essence of business? Do they enable managers to use proactive, rather than reactive, behavior and control practices? Do they cause managers to spend too much time finding problems rather than solving them?

An alternative view as shown in Exhibit 2 portrays the business body as being comprised of a collection of units called PODs. The name POD comes from a new approach to viewing business and is an acronym for *Preemptive Overview Design*.

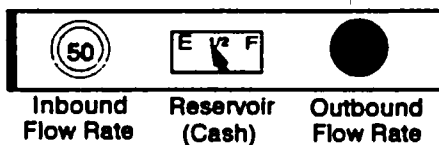
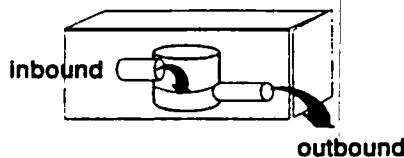
Each POD contains some combination of 4 basic elements: revenue centers, cost centers, reservoirs, and conduits. Revenue and cost centers are as traditionally defined in accounting. Reservoirs represent accumulations of revenues, expenses, and/or physical items such as inventory. Conduits represent flows of revenues, expenses and other items such as inventory into and out of reservoirs.

Exhibit 2. Collection of Business Units (PODs)



Conduits connect revenue centers, cost centers and reservoirs. The rate of flow through conduits affects the level of reservoirs. Similar to the flow of blood into and out of the human heart, income and expenses can be viewed as having characteristics such as flow volume and pressure. As techniques such as ultrasound are used to measure and portray bodily functions, business visualization techniques can be used to measure and portray business activities. Exhibit 3 illustrates a revenue center POD with revenue streams flowing through a conduit into a reservoir, and expense streams flowing through conduits out of that reservoir, thereby reducing the level of the reservoir.

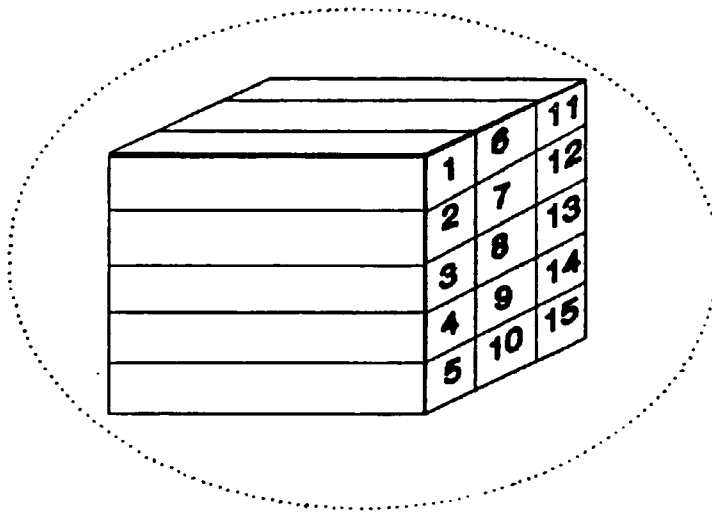
Exhibit 3. Revenue Center POD



Reservoirs and conduits as applied to business allow a more dynamic view of operations. Viewing business in terms of flows and reservoirs having different rates of being filled and depleted enables more effective and more timely control of business operations.

An overall view of the business, as represented in Exhibit 4, shows a collection of PODs enveloped in a spherical window. The spherical window is a representation of different views of the business such as those taken by strategic, tactical and/or operations managers. The sphere and the view it represents are omnidirectional, and can be used to examine any POD or combination of PODs at the discretion of the manager (Nielson, 1991).

Exhibit 4. POD Collection in Omnidirectional Spherical Window



Because PODs represent different functional and organizational units within the business, a strategic corporate-wide perspective might deal with companies, divisions, and regions. A tactical perspective might include stores and departments. An operational perspective might include departmental revenues, labor cost, materials cost, and overhead.

WHY BUSINESS VISUALIZATION

The operation of a business can be compared to the operation of an airplane. Imagine the outcome if we ran an airplane like a business. The pilot would be provided with a printed monthly report showing fuel consumption (the plane would run out of fuel before the pilot was made aware of the fuel level); a weekly report showing changes in altitude (the plane

could drop to below sea level in poor visibility before the pilot was made aware of the plane's altitude); and perhaps a daily report showing the aircraft's speed over the ground (the plane could easily overfly its destination before the pilot was made aware of the plane's location). Please don't book us on that airline!

The pilots of business are not always able to physically view all aspects of the business. They frequently rely on data and a limited number of gauges (graphs) (Barron, 1990). They are often unable to see where the business is going or where it has been. Unfortunately, traditional business instrument panels (financial statements and information reporting systems) are far less effective than those found in the cockpit of a 747.

Using the airplane example, it is neither new nor unusual for a large entity to be controlled either remotely or via surrogate devices. Patriot Missiles controlled by people and computers are able to shoot rapidly moving unseen projectiles out of the air. Large airplanes such as a 747 can be landed "blind" in storms with the aid of instruments and information systems.

Because the pilot cannot always see every part of the plane, he or she relies on surrogate images. These images are gauges and instruments that provide a continual stream of information about the plane's condition including speed of movement, changes in altitude, rates of change in altitude, and fuel consumption (Morrocco, 1991).

INSTRUMENTATION OF BUSINESS INFORMATION

It is time to consider the instrumentation of business information. Business instrumentation is defined as the use of computer generated gauges for the observation, measurement and control of the dynamic functions of a business. Business instrumentation should be considered now for the following reasons: First, advances in data acquisition devices such as point of sale cash registers, optical scanners (Perratore, 1991) and digital voice systems make it practical to capture not only traditional data but also images and sound. Second, data transfer capabilities such as Wide Area Networks (WANs), Local Area Networks (LANs), gateways, bridges, and inter-application data transfer (e.g., between word processing, spreadsheet, database, graphics and desktop publishing) simplify the movement of data (Stevenson, 1992). Third, the reduced cost of high powered microcomputer systems make hardware and software affordable (Pope, 1992). Fourth, expanded presentation options such as high resolution color screens, graphics software, color printers and multimedia programs offer practical visualization options that were not previously available. Fifth, increased user sophistication and the ever-increasing system performance expectations of users suggest that people can relate to the technology of business visualization.

Business visualization can enhance a manager's understanding of his or her particular business. Potential advantages include the reallocation of expensive management time from finding to solving problems; the refocusing of activities from reactive to proactive in such a way that revenues and costs can be influenced and controlled before they are incurred; the enhancement of quality; and doing more with limited resources, thereby increasing productivity and profit. In order to accomplish these tasks, it is necessary to identify certain basic constructs or axioms which provide the foundation for a business visualization system and then to develop appropriate implementation techniques.

BUSINESS VISUALIZATION AXIOMS

Business Visualization Axioms are the ideas that provide a framework within which BV systems and techniques can be applied. These include the following:

First is the axiom that *what can be measured can be managed*. Measurement implies that a benchmark, standard or reference point exists and efforts to change can be seen against the standard. For example, in service businesses labor productivity can be measured by the equation (work units produced)/(regular hours worked + 1.5 times overtime hours). This formula is a measure of labor productivity that describes the relationship between work produced and labor resources (manhours) used to produce the work. The constant 1.5 accounts for premium pay associated with overtime worked. Given this performance standard, goals can be set and productivity can be controlled.

Second is the axiom that *happenings and transactions can be given visual form*. Happenings and transactions are the everyday events of business. For example, the flow of money into cash registers in a grocery store checkout line can be recorded by electronic point of sale equipment. This data can be transmitted to a microcomputer and portrayed visually, in color and in motion using screen images such as water flowing through a pipe (our conduit concept). The rate of cash flow can be portrayed via the diameter of the pipe and/or the rate of motion through the pipe and can be measured dynamically by a gauge. Specific tools to accomplish this include new multimedia hardware and software.

Third is the axiom that *the visual form of happenings and transactions can be presented in a dynamic format* using current technology including animation and sound. At its root level this can be accomplished by applying mathematical engines to data such as income and expenses in order to calculate rates of change (Elliott, 1992), which can then be given visual form such as dollars per hour of sales on an "income gauge." Mathematical engines used to drive the gauges and instrument panels of business visualization can be tailored to reflect meaningful data collection, sampling and reporting frequencies.

Fourth is the axiom that *data and information need to be filtered* according to particular needs of the user. Managers at different levels in the firm have different information needs that derive from the same database (Clarke, 1992). In order for a manager to effectively see and control his or her area, it is necessary to screen out superfluous data. This is the process of filtering. In a large firm the sheer volume of data is overwhelming. Users having a "need to know" should be able to selectively filter out all data that is unnecessary to them, so they can readily see what is important.

Fifth is the axiom that *data can be focused*. The idea of focusing is that a user can see data at different levels of aggregation and can control the level of aggregation. For example zoom binoculars can be adjusted for vision at differing distances allowing the viewer to see different things. In a similar fashion, data can be viewed corporate wide, store wide, department wide, or at the item or individual transaction level to let the manager see different things.

Sixth is the axiom that *information should support proactive management action*. The case of a controller yelling at a manager for overspending last month's budget is a classic instance of reactive management. No amount of recrimination can change the outcome. The controller's information system should have warned him of the possible overrun long before it occurred so that he could have taken action before it was too late. Business visualization systems can provide immediate flashing screens and warning sounds to alert the controller to the problem.

These axioms provide the theoretical foundation for a business visualization system. The next step is to identify the technology components that can be used to build such a system.

DEVELOPMENT OF BUSINESS VISUALIZATION SYSTEMS

The primary purpose of a Business Visualization System is to make information more visible, more quickly in a more readily understandable format. This can be accomplished through the utilization of proper information system technology. Current information technology (IT), including hardware, operating systems, and application program development tools make it feasible to construct and deliver such a system.

The *hardware* for an optimum system platform is a powerful 486 or higher microcomputer running with a Graphical User Interface (GUI) program, such as WINDOWS 3.1¹, and linked electronically to data wherever it resides. The optimum data handling includes online collection devices (POS gear, electronic time clocks, optical scanners, voice mail systems) and linked to data bases.

The *software* includes programs called interfaces which are necessary for the transfer of data across various hardware platforms and between various databases; programs called Graphic User Interfaces (GUIs) which provide the basis for human interaction with the computer; programs such as computer languages and development environments which are used to construct the actual applications to be used by managers.

BV Systems can be created by using Visual Development Environments (VDEs) in conjunction with high level languages such as PASCAL and C. Visual Development Environments include Microsoft's Visual Basic, Asymetrix's Toolbook, KnowledgePro by Knowledge Garden, Inc., and Borland's ObjectVision (Petzold, 1992). Although VDEs utilize different techniques, require different levels of programming skills, and include different features they share a common goal of creating applications to run in the Microsoft Windows Environment.

The availability of these hardware and software platforms make it possible to create a dynamic business visualization system.

¹WINDOWS 3.1 is a Registered Trademark of Microsoft Corporation.

COMPONENTS OF BUSINESS VISUALIZATION SYSTEMS

Primary components of BV systems are filter tools, view tools, item tools, time tools, and projection tools. Each of these tools has a specific purpose within the Business Visualization System.

The Filter Tool separates critical data from the mass of available data. Managers have need to see particular data in order to effectively control their operations. For example a firm's database may include labor cost and revenue or sales information. A manager may be concerned only with labor data. He may further be concerned with only a particular department or work area. He may also be concerned with a particular day or work shift. Filter tools enable him to select the relevant data thereby removing or filtering out that which is not critical.

The View Tool is used to visualize filtered data. Existing systems limit the way in which data can be viewed. For example a grocery store manager may wish to see sales processed by each cashier. Presently this can be viewed as a table of data showing levels of sales transactions at fixed intervals. In contrast a view tool can portray the sales as water moving through a pipe or conduit having an associated gauge showing the rate of cash flow. The view tool can simultaneously display multiple cashiers in this manner thereby enabling the manager to see in a very dynamic fashion the relative behavior of the overall cashiering activity. Such a view tool creates an easy to understand pictorial environment.

The Item Tool provides access to filtered information related to a specific product or item. It is similar to the filter tool in that it enables a manager to select specific information to be viewed. It differs in that it enables the selection of product/item data in a way that can show the relative behavior of similar items. For example in a grocery store it is useful to compare sales of different brands, styles and sizes of peanut butter. The item tool enables the manager to select the category of item (peanut butter), the brand of item (Jif, Peter Pan, Skippy), the style (creamy or chunky) and the size (16 oz, 24 oz, 32 oz). The item tool is used to measure the response of the item to business patterns, management actions and market conditions.

The Time Tool enables the user to select relevant time periods for analysis of filtered data. A time tool is a calendar superimposed over a database in such a way that the user can select the time horizon or horizons that are meaningful and relevant. For example a hotel manager may need to see the room sales behavior of Tuesdays for the month of November as compared to that of the preceding month or the same month of the preceding year. The time tool allows the manager to mix and match time periods in a very flexible manner.

The Projection Tool enables users to view possible future outcomes based on known data. Although sophisticated forecasting tools and techniques are currently available, they are used primarily by statistical experts. The projection tool component of a business visualization system is intended for use by managers who may have limited statistical training, but know what it is that they need to project. For example in the middle of the third week of a month, a manager may wish to know his approximate labor cost for that entire month. The projection tool can use actual labor hours and costs based on electronic time clock information for the first and second weeks. The third week can be projected by adding actual hours and costs for the first 4 days to the scheduled hours for the remaining 3 days. The last week of the month can be taken from the budget. Summing the results of these will provide a projection of month-end labor cost.

The preceding is one example of how a projection can be made. The projection tool includes a collection of preprogrammed forecasting techniques which can be easily applied to the tasks at hand. Techniques include forecasting, extrapolation, simulation and expert systems.

When filter, view, item, time and projection tools are combined in a user friendly environment the result is a very powerful and easy to use business visualization system.

LOOKING AHEAD

The changing business and technological environment has led to a need for a better way to view business operations. In order for business visualization to succeed it is necessary to move from the mindset of making pretty that which is static to dynamically representing business operations in such a way that business managers are able to pilot their business as it occurs in a proactive rather than reactive manner.

Using Business Visualization Systems in the workplace will require changes in attitude, behavior, and information systems (Stewart, 1992). To move in this direction it is necessary to recognize the opportunity, to clearly define expectations of new, dynamic measures of business activity, and to construct such a model. The technology is available and software is in place. All that is needed is a commitment and willingness to embrace the new approach to Business Visualization.

As a note of caution, the approach described in this article can be both powerful and dangerous. It is dangerous because, taken to extremes, the BV systems could result in the embodiment of George Orwell's 1984. It is powerful in an organizational sense because it can enable people close to the production process to better align resources with needs (Lawler, 1991). This supports the concepts of Total Quality Management.

REFERENCES

- Barron, D. (1990, May 15). Ease of use coming to PC graphics. *DATAMATION*, pp. 37-42.
- Clarke, M. K. (1992, Winter). Increasing the competitiveness of America's manufacturers: A review of state industrial extension programs. *Economic Development Review*, 10(1), 88-89.
- Elliott, R. K. (1992, June). The third wave breaks on the shores of accounting. *ACCOUNTING HORIZONS*, 6(2), 61-85.
- Lawler, E. E., III. (1991, Summer). The new plant approach: A second generation approach. *Organizational Dynamics*, 20(1), 4-14.
- Marcus, A. & van Dam, A. (1991, September). User-interface developments for the nineties. *IEEE JOURNAL*, 49-57.
- Morocco, J. D. (1991, March 25). McDonnell F/A-18E/F offers increased range, endurance. *Aviation Week & Space Technology*, pp. 25-26.

- Nielson, G. M. (1991, September). Visualization in scientific and engineering computation. *IEEE JOURNAL*, 58-66.
- Perratore, E. (1991, November 12). Quick graphics on a budget: 12 hand scanners. *PC Magazine*, 10(19), 223-281.
- Petzold, C. (1992, June 16). The visual development environment more than just a pretty face? *PC Magazine*, 11(11), 195-237.
- Pope, K. (1992, October 20). Compaq slashes prices on computers in possible new round of PC battle. *The Wall Street Journal*, p. B3.
- Stevenson, T. (1992, September 15). Low-cost integrated software: The new synergists. *PC Magazine*, 11(15), 311-364.
- Stewart, T. A. (1992, May 18). The search for the organization of tomorrow. *Fortune*, 125(10), 92-98.
- Valovic, T. (1992, June). Can high-speed networks help US companies regain a competitive edge? *Telecommunications* (North American Edition), 26(6), 25-28.