A Framework for Comparing Information Engineering Methods

By: Richard D. Hackathorn
Associate Professor
of Information Systems

Jahangir Karimi
Assistant Professor
of Information Systems

College of Business and
Administration
University of Colorado at Denver
1475 Lawrence Street
Denver, Colorado 80202

Abstract

This article proposes a comprehensive framework for comparing the current methodologies and tools for information engineering and using these methods for applying information technology to construct the overall information systems architecture for the organization. The proposed framework consists of two dimensions: (1) an expanded sequence of the traditional system life cycle; and (2) the conceptual depth of the methods.

The article shows that information engineering is the key to effective information management. Using the proposed framework, 26 widely-cited methods for information engineering are compared. Evolution to more effective methods of information engineering are needed to align future information systems requirements to strategic goals and objectives of an organization and to exploit the current information systems technologies for competitive advantage.

Keywords: Information engineering, information systems architecture, information systems design, software engineering, system development methodologies

ACM Categories: K.6.1, K.6.3, H.1.1

Introduction

This article proposes a comprehensive framework for comparing the current methodologies and tools for applying information engineering to construct the information systems architecture (ISA) for an organization. Without a solid grasp of how the organizational goals are translated into an overall architecture, an organization pursues the development of information systems according to a short-term perspective of isolated applications. Thus, a fragmented information system results that hinders the ability of the organization to respond to changing environments.

What is information engineering?

The term *information engineering* (IE) was coined by Martin (1982), who describes it as "data that is stored and maintained by computers and the information distilled from data" (p.15). In this article the term is used more broadly.

Information engineering is directed specifically at translating a corporate focus (a strategic plan, expressed as organizational mission statement) into an *information systems architecture* (ISA), which can be directly translated into data, application, and geographic architectures. The idea of engineering is used because, with data recognized as a corporate resource, an analyst must work hard to exploit its value (i.e., to "engineer" for its multiple uses). To engineer information, the analyst needs to know and understand the organization, or more specifically the functional activities associated with the organization's business system (Jackson, 1986).

Bryan, et al. (1982) defines software as information that is: (a) structured with logical and functional properties, (b) created and maintained in various forms and representations during its life cycle, and (c) tailored for machine processing in its fully developed state. The fundamental purpose of *software engineering* (SE) is also to apply engineering principles to the software life cycle. This involves a series of well-defined steps that comprise the essential elements of software planning, development, and support.

The fundamental difference between IE and SE is in their breadth of systems life cycle activities. The information engineer begins the process by translating the organizational strategic plan into the ISA using a set of IE methods. The software engineer continues where the information engineer leaves off, namely, at the composite logical model of the organizational information systems requirements as they pertain to the ISA. This logical model specifies the requirements for software and hardware to support the ISA. The SE methods can be employed to plan, design, and implement the software component of the overall architecture.

The benefits associated with orchestrated use of IE methods are the same as those resulting from the architected information systems environment. The stability of systems is the primary goal of information engineering (Inmon, 1986). Stability is achieved when the systems are built based on the ISA. An architected information systems environment provides a framework so that one development effort can be built on another. In an unmodeled environment, growth is unplanned, overlapping, and generally disorganized. The more growth there is, the more difficult the environment is to manage.

Further, an architected information systems environment serves as a blue print on which systems development activities can be: (a) prioritized by deciding the sequencing for building the architecture; (b) coordinated by relating to other systems in an efficient manner; and (c) optimized by building each system with the corporate information requirements in mind (Wetherbe and Davis, 1983).

Need for information engineering

Systems development activity has been a "bottom-up" activity in which various functions and data areas are automated on an application-by-application basis without great consideration for integration and optimization at the organizational level. Organizations were satisfied with this situation until recently, when many of them discovered that the automated pieces were increasingly interdependent, incompatible, redundant, and incomprehensible.

Therefore, a critical step for any large organization today is to establish ongoing planning and development activities for information engineering. Such activities should direct the architecture of information systems and satisfy diverse and changing needs across the entire organization.

In general, IE activities are difficult to perform because of the enormous amount of highly detailed requirements specifications, mixed with vague statements of organizational objectives. The clash of the bottom-up synthesis of these requirements with the top-down decomposition of objectives often results in confusion among the responsible parties, rather than a coherent plan of action (Inmon, 1986).

Further, there is often a failure to communicate properly between those with a "business" perspective and those with a "technology" perspective — two groups that must form a close working team to survive in many of today's industries.

There are many competing methods for performing information engineering, each having its unique strengths and weaknesses. For example, most methods have focused on the bottom-top and technically oriented aspects, thus appearing at times to be irrelevant to top management. The few methods with a strategic orientation can consume major resources and produce few tangible results for computer systems design.

Within the industry there is considerable attention currently focused on IE methods, with few prospects of resolution in the near future. Because of the competitive nature of this marketplace, comparative evaluation of existing methods is lacking, and objective statements for future directions are proprietary.

Relationship with prior research

Systems that were originally computerized were limited in scope and involved automation of manual processes or procedures. This resulted in a process view of systems development that endured for many years. The focus changed with the realization that processes were subject to constant changes and modifications. As a result, focus shifted to the collection of data used by a process, rather than upon the process itself.

Eventually data was viewed in an organizational context, and it became necessary to develop tools that would reflect this. Recently there has been important recognition that the focus of design should be on data pertaining to business entities, rather than business processes. As a result, the focus of the systems development methods has also been shifted towards information systems requirements determination as it relates to global organizational strategic goals and objectives.

Spanning over thirty years, there is an immense body of literature dealing with traditional systems development methodologies. This article will refer to these methodologies as systems development life cycle (SDLC) methods. Reviews of the SDLC literature (Colter, 1984; Couger, et al., 1982) classify the methods into generations in order to highlight the evolution over time. The objective of these methods is to develop an application (i.e., computer-aided business function) embodied within an information systems. The process is divided into phases and managed on a project basis. Typical phases are:

- Feasibility study
- Systems analysis
- General systems design
- Detailed systems design
- Systems implementation
- Operation
- Maintenance and evaluation

Through the 1970s, various "structured" methods emerged (Colter, 1982). Although structured methods offered improved analysis and design capabilities, no single method has dominated other methods in its comprehensiveness. As noted by Colter (1982), the structured methods have failed to provide a coherent approach to automatic design of high quality systems.

The above research made progress on *how* to develop applications; however, effective determination of *what* applications to develop is lacking. Organizations in the industry each dealt with the "what" issue by establishing a steering committee that had company-wide representation and commitment from top management. This steering committee gathered ideas for applications from others, set priorities, and allocated development resources. The steering committee provided a political solution to the critical allocation issue concerning the infamous "application backlog."

In the next section, an analytical framework consisting of two dimensions — breadth and depth — is proposed for comparing IE methods.[1] This framework suggests roles for planners and developers and two processes — align and exploit — that assure that organizational goals and information systems architecture are compatible.

[1] This article will extensively use the term *method* as a generic term that refers to any combination of methodologies, techniques, and tools used for IE.

The research approach for selection and classification of sets of IE methods to support IE activities will be discussed in the last two sections. In particular, these sections will describe a selected set of IE methods and their implications to the organization's strategic planning for information systems.

Analytical Framework

To compare the various IE methods, an analytical framework consisting of two dimensions was used. These dimensions are named *breadth* and *depth* and form the axes of a graph (called the DB-space) that compares the IE methods.

The breadth dimension

The breadth dimension of the analytical framework is an extension of traditional frameworks for the systems development life cycle in which the overall mission and nature of the organization are included. A traditional systems development life cycle begins with an examination of the feasibility (i.e., cost and benefits) of developing an application. Usually omitted are considerations of the strategic implications to the organization and the global architecture for all current and future applications. Hence, the breadth dimension deals first with the strategic consideration of information management and then with the tactical and operational details of information systems.

The breadth dimension has five phases (see Figure 1). They are:

- Organizational analysis
- Strategy-to-requirement transformation
- Logical systems design
- Logical-to-physical transformation
- Systems implementation

In Figure 1 the rectangles denote activities that are performed. The circles show products from the preceding activity that are used in the subsequent activity. Thus, the breadth dimension describes both what is being done and what will result. Each activity followed by its product is referred to as a phase.

Note that information engineering (as defined in the first section) focuses on the first half of this figure and results in the information systems architecture. Software engineering, on the other hand, focuses on the latter half.

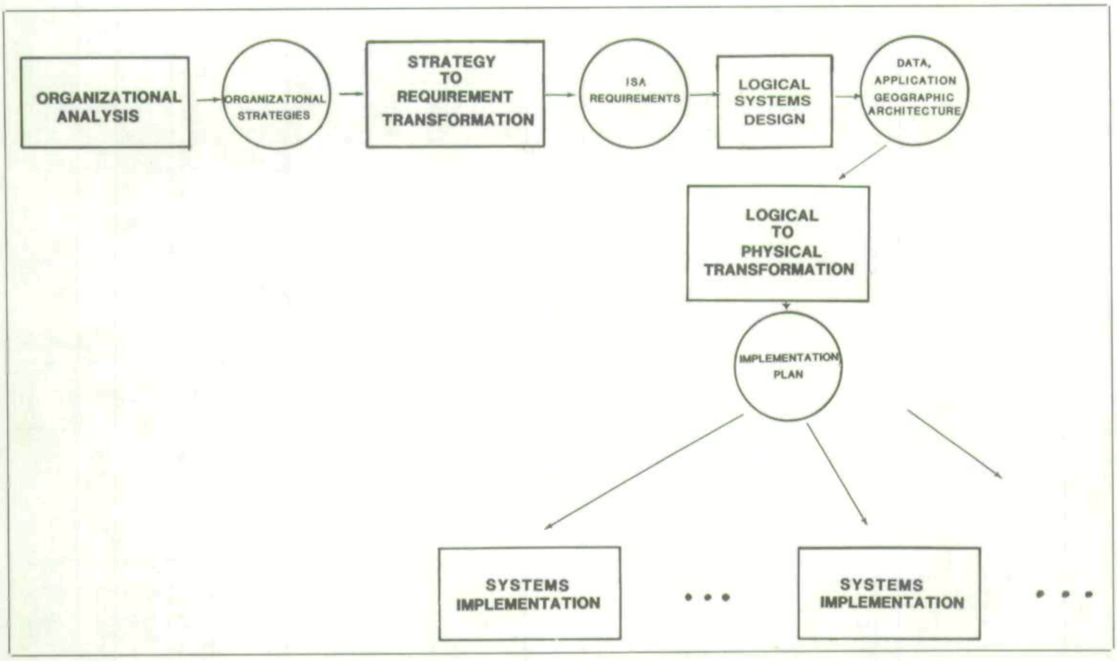

Figure 1. Phases of the Breadth Dimension

Also note that the first four phases are global to the organization, while the last phase — systems implementation — is local to a specific application. Some activities, such as logical design, will be replicated within systems implementation but at a finer level of detail oriented to a specific application.

The remainder of this section describes each of the five phases in more detail.

Organizational Analysis

The first phase of the breadth dimension is *organizational analysis.* The first purpose of organizational analysis is to examine the mission and nature of the organization and its environment. The second purpose is to translate these aspects into a set of organizational strategies. The intent is to produce a concise, accurate, and formal statement of the organizational strategies that will be useful in the second phase.

Organizational analysis is increasingly important to the proper development of an information system. The dynamic nature of today's organizations, whether private or public, requires a continual reassessment of their information resources and the proper management of these resources.

Strategy-to-Requirement Transformation

The second phase of the breadth dimension is *strategy-to-requirement transformation.* Its purpose is to model the information systems architecture (ISA) that represents the information flow requirements of the entire organization. Organizational strategies are used for logical modeling of the ISA.

The ISA relates the organizational processes that must be performed to data classes that are required by those processes. This architecture represents the information flow requirements of the entire organization, and it is further refined to data, application, and geographic architectures for discrete organizational units (Inmon, 1986; Schouw, 1983; Wardle, 1984). Modeling for the ISA consists of the following activities:

— Global entity relation modeling
— Conceptual data modeling
— Process modeling
— Data/process integration

It includes the following subactivities:

— Problem scope and definition
— System boundaries
— Requirement specification
— Integration scope

— Procedures for carrying out the tasks
— Frequency of task
— Task-to-document usage
— View identification and integration

The logical modeling for the ISA, at minimum, is to (1) include the needs of all users of data processing services, and (2) minimize redundancy in data and process modeling across the organization. The ISA that results is used as the basis for performing logical systems design and establishing commitment for systems implementation.

Logical Systems Design

The third phase of the breadth dimension is *logical systems design.* The purpose of this phase is to design data, application, and geographic architectures using the logical model of the ISA.

The data architecture represents a blueprint of the databases that should be designed from an organizational standpoint. The application architecture defines the application areas necessary to support the ISA and the relationships between those applications. The geographic architecture describes where applications will run, where databases will be located, and what communication links are needed between the locations.

The logical design for these architectures consists of the following components:

— Entity/relation diagrams (ERD) and global ERD
— Complete, consistent semantic data modeling including:
  — entities (objects)
  — properties
  — relationship (is-a, is-part-of)
  — functional dependencies
  — events and actions
  — data item set (keys)
— Directories and dictionaries
— Process model
— Operational mode
— Dialogue and communications procedures

Logical-to-Physical Transformation

The fourth phase of the breadth dimension is *logical-to-physical transformation.* This phase consists of decomposing data, application, and geographic architectures, into subsystems (or portfolio of applications), deciding on the detailed design of each subsystem, and making commitments to prioritize and schedule subsystems implementation.

The result is the detailed systems design *implementation plan* that describes projects (i.e., the steps to implement a specific subsystem or application). The detailed systems design implementation plan is the basis for directing the implementation of a set of systems. In general, this product includes the following:

— Schema and subschemas specification of databases
— Software specification
— System components specification

Systems Implementation

The final phase of the breadth dimension is *systems implementation.* This phase occurs many times — once for each system defined in the detailed systems design implementation plan. The result is an operational subsystem that supports a business function of the organization. This phase is similar to the traditional systems life cycle that is initiated with a feasibility study of an application. The exception is that systems implementation should start with a notion of the ISA, guiding the implementation in its integration with other subsystems.

The depth dimension

The depth dimension of the analytical framework deals with the conceptual-to-practical dimension of an IE method. On the conceptual side of this dimension, a method should be a solid basis for explaining its approach, major issues, relationships among variables, and expected outcomes. Having such a conceptual basis, the method has more consistency and stability in the industry. On the other hand, the practical side of the depth dimension focuses on tools for actually performing the method, considering issues of useability and efficiency.

The depth dimension consists of three levels: methodology, technique, and tool. As mentioned above, *method* refers to any combination of these levels.

The term *methodology* is defined as "the analysis of the principles. . .of inquiry in a particular field" (*Webster New World Dictionary,* 1981). This definition emphasizes the conceptual basis for performing the IE activities (i.e., the "what") and highlighting questions like the following:

— What factors or variables are important?
— What are the relationships among these factors?
— What are the desirable outcomes?
— What management actions can be taken from these outcomes?

The term *technique* is defined as "a procedure for accomplishing a desired outcome." In particular, a technique specifies the steps in performing the IE activities, as well as the necessary inputs and results from each step. A technique deals with the logical way of "how" to do an activity and represents knowledge more than actual products.

The term *tool* is defined as "an instrument for performing a procedure." In particular, a tool is some tangible aid (e.g., analysis form or computer-assisted software program) used in performing some aspect of IE. The objective of using a tool is to produce a deliverable.

As an example of these three levels, consider structured design (Yourdon and Constantine, 1979). The principles of the structured design are devoted to attaining modules that have three important properties: (a) modules are relatively independent; (b) existing dependencies can be easily understood; and (c) there are no hidden or unforeseen interactions between modules. Structured design also provides a set of techniques for attaining modules with the above properties. A set of guidelines is also provided for distinguishing a "poor" and a "good" design. However, the structured design does not provide automated tools for creating a system of modules with the specified properties from the system requirement specification. Therefore, the structured design covers an area within the methodology and technique range, as will be explained further in the section comparing IE methods.

The purpose of the depth dimension with its three levels is to view the various IE methods as they relate to both their conceptual foundations and practical results.

DB-space

The analytical framework covers an inclusive systems development life cycle starting with organizational analysis and ending with systems implementation. Further, the depth to which each IE method spans the levels of methodologies, techniques, and tools has been noted.

The two dimensions of breadth and depth are used together to form a framework for comparing the various IE methods. A graph was constructed using the breadth dimension as the horizontal axis and the depth dimension as the vertical axis. The space (four quadrants) formed by the two dimensions is referred to as the *DB-space* and is used to characterize each IE method.

Planner and developer roles

A useful and simple explanation of the DB-space is provided by considering the nature of the four quadrants, for example, by considering these four quadrants as *roles* that can be performed within ongoing IE activities of the organization. As shown in Figure 2, the four quadrants are labelled counterclockwise from the upper left as:

— Conceptual Planner
— Pragmatic Planner
— Pragmatic Developer
— Conceptual Developer

Conceptual planners are concerned with (1) the organization's strategic planning and direction setting and (2) the establishment of a corporate policy for the key technologies (i.e., information systems) to gain competitive advantages in the market place. Hence, paradigms or frameworks of organizational analysis are their focus.

Pragmatic planners are concerned with modeling the organization's structure, policies, and procedures and investment strategies and with using these models to drive the information systems requirements.

Pragmatic developers are concerned with implementing the information systems architecture. They are not directly concerned with long-range organizational policy and objectives and ignore the abstract eloquence of the system.

Conceptual developers are concerned about the conceptual basis for employing technology to meet the goals and objectives of the organization, and setting new direction.

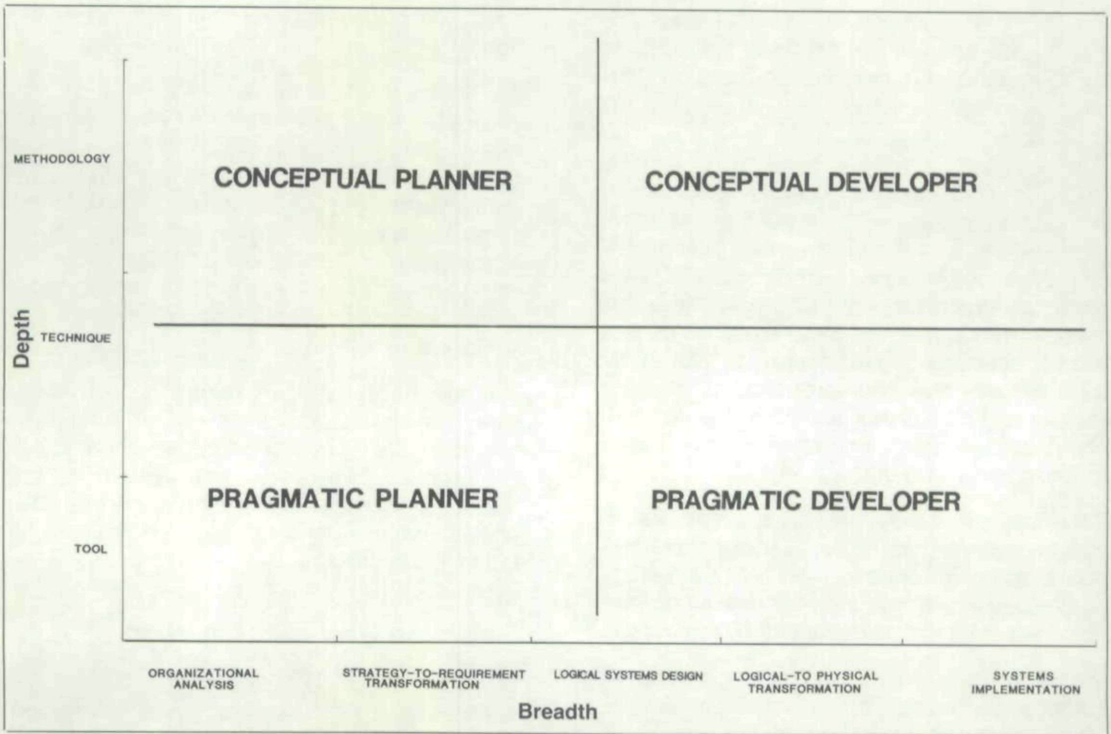

Figure 2. Four Roles of Planner and Developer

Research Method

The intent of this study has been: (a) to determine the general nature of a framework for characteristics of IE methods necessary for comparing various IE methods; (b) to systematize the enumeration of the characteristics framework; (c) to select and classify a representative set of IE methods in order to populate the framework for comparing IE methods with some examples sufficient to demonstrate its effectiveness and rationale; and (d) to show how the authors' analytical framework can be a good tool for comparing various IE methods if such a comparison should take place within a development environment.

This study does *not* intend to evaluate the limitations or potential benefits associated with each of the IE methods since this evaluation is dependent on a number of factors such as: (a) the current state of the information technology employed; (b) the level of integration between the methods; (c) the perceived suitability of the method selected in specific applications; (d) the experience with using the method in the organization, including the required efforts to introduce the method, the extent of projects included, and the organization involved; (e) the skills and the expertise required in using the methods; and (f) commercial support, pricing, training, etc. involved in purchasing the software package. Also, a general statement regarding potential benefits and constraints of an IE method is not likely to be regarded as useful since the usefulness of an IE method is very much dependent on a number of factors (including the ones just mentioned).

The following sections describe the characteristics framework necessary for selection and classification of IE methods. Then, specific IE methods are selected and placed on the DB-space, and conclusions are drawn about the coverage and evolution of the methods.

Method characteristics framework

The characteristics of the methods are central to the development of a methodology for comparing various IE methods. Potentially, there is a very large number of characteristics useful for comparing IE methods. These characteristics may be attributes of: (a) products produced by the IE method; (b) the IE method itself; (c) developers of the IE method; and (d) users of the IE method.

The parameters useful for comparing various IE methods are based on the characteristics of the methods. The set of characteristics may potentially continue to grow to reflect consideration of new parameters for comparison. A framework is introduced below for the characteristics that can be extended as necessary and can easily accommodate new characteristics as they are uncovered. To be useful in defining the parameters, ill-defined characteristics need to be based on those more well-defined.

The overall structure for the characteristics framework is presented in Figure 3. These characteristics highlight major concerns which arise when: (a) considering IE methods for use on IE projects; and (b) comparing various IE methods.

The parameters used for comparing various IE methods are based on two major characteristics: (a) extent of coverage over the breadth dimension; and (b) extent of coverage over the depth dimension. Extent of coverage is the degree to which an IE method addresses the parameters of major concerns (i.e., breadth and depth) on the input or the output side. These two parameters are defined below.

The first parameter is the extent to which an IE method covers the ISA development process. The input to the method may be generated during any of the five phases of the breadth dimension. The output produced can be useful in the same phase and/or any of the subsequent phases of the breadth dimension. Figure 3 shows that a method having its required input generated in the logical systems design phase has potential to be useful in the same phase and/or in the two remaining phases (i.e., logical-to-physical transformation and systems implementation).

The second parameter is the extent to which an IE method disciplines or directs the creation and need evaluation for the ISA. The form of the analysis employed by the method and the form of the output generated by the method are addressed in the depth dimension, and it may differ between methods. The form of the analysis and/or description provided can be in part or in whole conceptual (methodology), procedural (technique), and/or machine processable (tool).

The rating involves three levels: none or little coverage, partial coverage, and extensive coverage. The lowest level means that the method does not address the dimension of interest. The partial rating is given to methods that provide some but incomplete coverage of the dimensions (i.e., $0 < \text{Coverage} \leq 50$). The extensive rating is given to methods that have strong capability to address the dimension of interest (i.e., $50 < \text{Coverage} \leq 100$).

This framework can be extended to accommodate new characteristics if needed. Potentially, new characteristics can be added anywhere in the structure. However, it is expected that the two major characteristics will be relatively static, and the future extensions will lead to additions to these parameters.

Selection and classification of IE methods

After enumerating the characteristics framework, a two-stage process consisting of selection and classification was used for comparing various IE methods. This approach is similar to the one taken in STARS software methodology selection (McDonald, et al., 1986) and has the benefit of highlighting relationships and dependencies among the IE methods.

The first stage for comparing IE methods was *selection* of the methods and was based on the following criteria: (a) to be representative of coverage over the entire analytical framework; (b) to illustrate any overlap or gap among the existing IE methods; (c) to illustrate trends in the evolution of IE methods; and (d) to highlight the relationships among the IE methods for facilitating and balancing each other.

All existing IE methods were not considered in this analysis. Rather, the selection process focused on candidate IE methods and chose among the candidates. Each of the twenty-six methods selected was described in detail in the full report of this research (Karimi, et al., 1985).

The second stage was *classification*, which determined the characteristics of the IE method based on the parameters of major concern (defined above) and classified them (to place them on the DB-space in the proper cell) based on the determined characteristics.

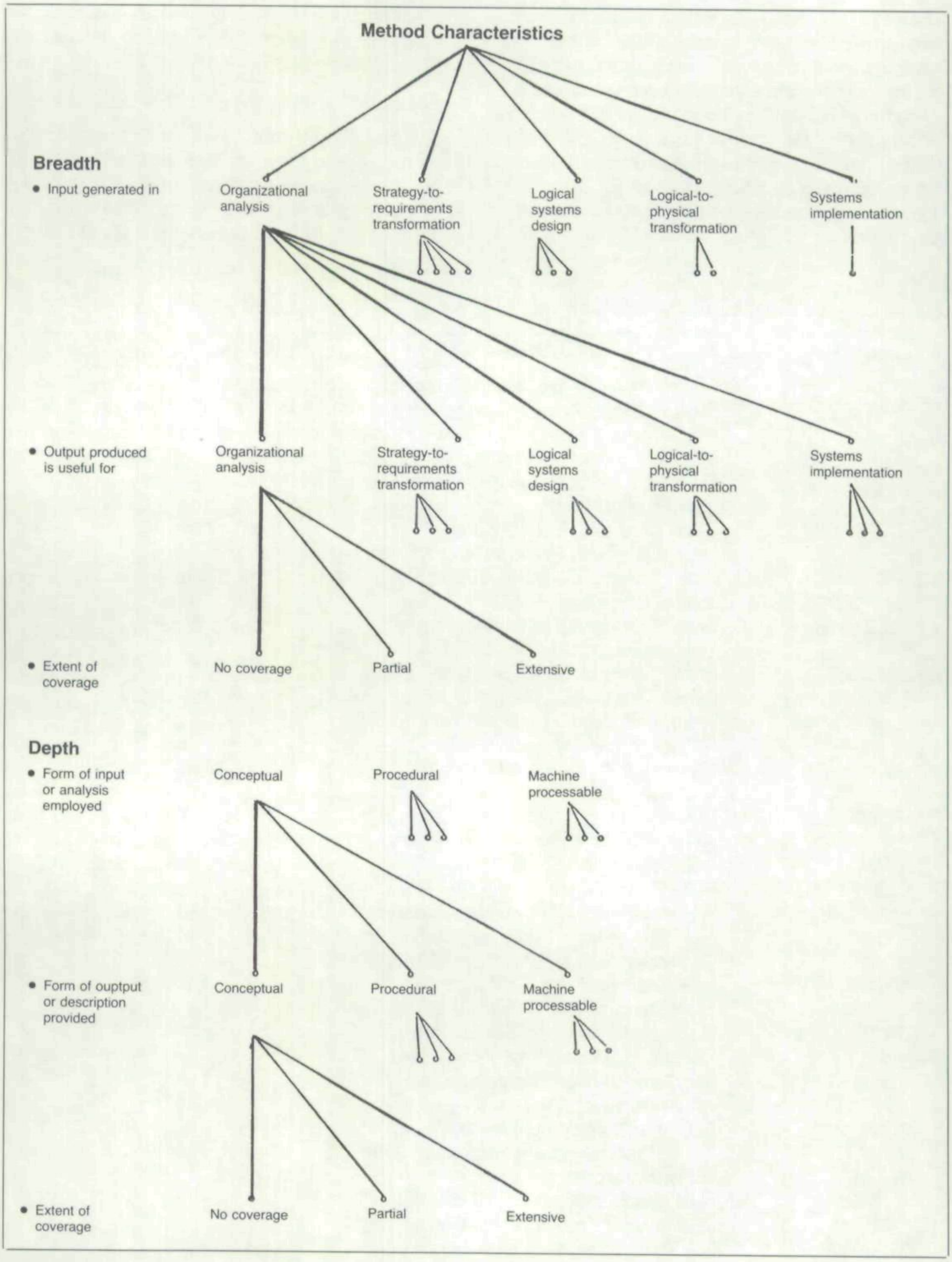

Figure 3. Method Characteristics Framework

The selection and classification process evolved through iterations. At each iteration consideration was given to: (a) focussing upon candidate IE methods and choosing among the candidates using the previously developed selection and classification process as necessary; (b) using the method characteristics framework to determine the method's characteristics; and (c) using the analytical framework to define the classification for a method.

A listing of the various IE methods selected with their full names and vendors (for commercial products) is given in Figure 4.

Results and discussion

The process for classifying the IE methods was to: (a) review the literature on the IE methods selected; (b) contact the vendors (in the case of commercial products) for additional information; (c) talk to local users of the product about

Label	Name	Vendor
ACG	Automatic Code Generation (Blosser, 1975)	
BIAIT	Business Info. Analy. & Integration Tech. (Carlson, 1979)	
BSP	Business Systems Planning (IBM, 1975)	IBM
CAPO	Computer-Aided Process Organization (Karimi, 1986-87)	
CASE 2000	Case 2000 (Nastec, 1985)	Nastec Corp.
CSF	Critical Success Factors (Rockart, 1982)	
D-D	Data Designer (Database Design Inc., 1981)	Data Designer
DATA DICT	Data Dictionaries	
DSSD	Data Structured System Development (Warnier, 1981)	Ken Orr & Assoc.
E-R	Entity-Relationship Model (Chen, 1976)	
EXCEL	Excelerator (Index Technology Corp., 1986)	Index Technology
EWIM	Enterprise-Wide Info. Management (Benson and Parker, 1985)	
ISA	Information Systems Architecture (Inmon, 1986)	Amer. Mgt. Systems
ISDOS	Info. Sys. Design and Optimization (Teichroew and Hershey, 1977)	ISDOS, Inc.
M-W	Methodware (Appleton, 1985)	D. Appleton
PLEXPLAN	PLEXPLAN (McIntyre, 1986)	
PSL/PSA	Problem Statement Lang. & Analyzer (Teichroew and Hershey, 1977)	ISDOS, Inc.
RDM	Relational Data Model (Codd, 1970)	
SADT	Structured Analysis & Design Technique (Ross, 1985)	
SAST	Strategic Assumption Surfacing & Testing (Mason and Mitroff, 1987)	
SD	Structured Design (Yourdon and Constantine, 1979)	
SDM	Systems Development Methodology (AGS Management Systems, 1985)	AGS Mgt. Systems
SREM	Software Requirement Eng. Methodology (Alford, 1985)	TRW
SST	Strategy Set Transformation (King, 1978)	
STRADIS	Str. Anal., Design, & Implementation of IS (Gane, 1984)	McAuto/IST
TAXIS	TAXIS (Mylopoulas, et al., 1980)	

Figure 4. Listing of Methods

their experiences; and finally (d) rate the coverage of each method in comparison to the outcome expected from each phase of the ISA development life cycle as discussed previously.

For each method, the extent of coverage with respect to breadth and depth was then rated by performing the following four tests (i.e., two tests for each of the two dimensions):

1. The input to the method requires (none, partial, extensive) knowledge about the (organizational analysis, strategy-to-requirement transformation, logical systems design, logical-to-physical transformation, systems implementation) phase of the ISA development life cycle.
2. The output produced from the method provides (none, partial, extensive) knowledge useful for the (organizational analysis, strategy-to-requirement transformation, logical systems design, logical-to-physical transformation, logical systems design, logical-to-physical transformation, systems implementation) phase of the ISA development life cycle.
3. The input to the method requires (none, partial, extensive) (conceptual, procedural detail, machine processable) knowledge.
4. The output produced from the method provides (none, partial, extensive) (conceptual, procedural detail, machine processable) knowledge.

These ratings were performed subjectively by the authors according to the process described above and based on their best knowledge as gained from the information collected and from personal experience. While judgmental, these ratings were supported by the literature on each method and by contacts with the vendors and, in some instances, with users of the commercial products. Figure 5 shows the results of this rating process.

Coverage of IE Methods According to Breadth and Depth

	OA		S-RT		LSD		L-PT		SI		METH		TECH		TOOL	
	I	O	I	O	I	O	I	O	I	O	I	O	I	O	I	O
ACG									P	P			P	P	E	E
BIAIT			E	E									E	E		
BSP	P	P	E	E								P	E	E		
CAPO							P	P					P	P	P	P
CASE 2000							E	E	E	E					E	E
CSF			E	E							P	P				
D-D					E	E	P	P					E	E	E	E
DATA DICT									P	P					E	E
DSSD							P	P	P	P			E	E	P	P
E-R					P	P					P	P	P	P		
EXCEL					P	P	E	E	E	E					E	E
EWIM	P	P	P	P							P	P				
ISA			E	E	E	E					P	P	E	E		
ISDOS					E	E	E	E	E	E			P	P	P	P
M-W					P	P	P	P					P	P	P	P
PLEXPLAN			E	E									P	P	E	E
PSL/PSA					E	E	P						P	P	E	E
RDM					P	P						P	P	P	E	E
SADT					E	E							E	E	P	P
SAST			P	P	E	E					P	P	E	E		
SD							E	E			P	P	E	E		
SDM									P	P			P	P	E	E
SREM					E	E	P	P					P	P	E	E
SST	P	P	E	E							E	E				
STRADIS					P	P	E	E	E	E			P	P	E	E
TAXIS					P	P					P	P	E	E	P	P

Figure 5. Extent of Breadth and Depth Coverage

In addition to judgmental ratings, there are several alternative ways to determine these ratings. For instance, a panel of experts could generate ratings which would indicate variance among the experts for each method. A field survey would compile the perceptions of interested professionals. Finally, ratings of greater validity could be produced via laboratory experiments using simple cases that focus on specific portions of the breadth and depth dimensions. The laboratory alternative, however, could only deal with a limited number of methods.

The results in Figure 5 were used to derive the comparison of methods in Figure 6. The size and placement of the box representing each method in Figure 6 indicates the relative extent of coverage for the breadth and depth dimensions. It is the reason for considering that the partial and extensive ratings indicate ranges of coverage (i.e., $0 < C \leqslant 50$ or $50 < C \leqslant 100$); therefore, the size and placement of the boxes representing any two methods may vary slightly even though the two methods may have the same ratings in either dimension.

As an illustration of the process employed for estimating the size and placement of the box representing each method, consider the box representing structured design (SD). SD was characterized earlier as a combination of methodology and technique, but was not a tool. Yourdon and Constantine (1979) describe SD as useful in the development of nonprocedural specifications for the modules within a software system. These specifications are related to all module interconnections and module functions. SD also provides a set of properties and a set of techniques for creating software with the desired properties. However, the success of design using these techniques relies upon the designer's self discipline and professional judgment to ensure that design decisions are not based on speculation or premature selection of alternatives.

Figure 6 shows that the box representing SD covers an area within the methodology and technique range (in depth), and an area within the logical-to-physical transformation range (in breadth). However, the specification for software is not the only outcome of this phase of ISA development; specifications for the schema of databases and other system components such as hardware and telecommunication network are outcomes as well.

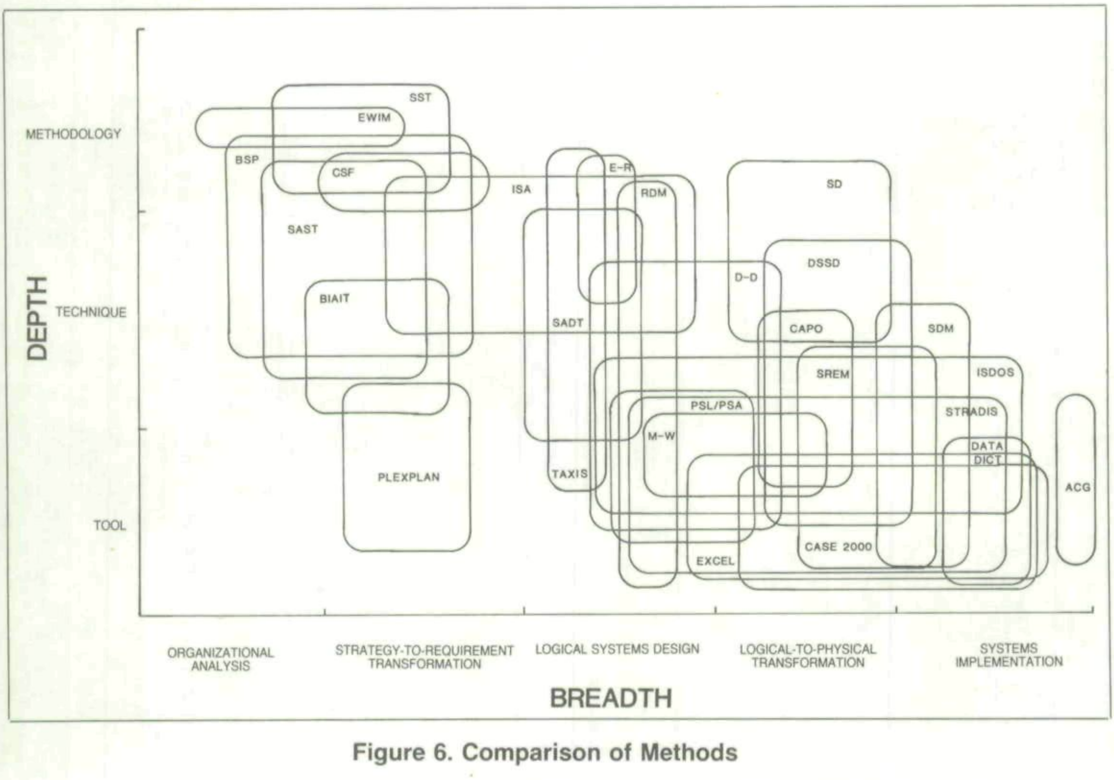

Figure 6. Comparison of Methods

Although the framework does not indicate which of the various IE methods classified within a phase should be used for that phase, the framework does help prevent an organization from using a system implementation method when an ISA modeling method is needed. The classification is not meant to imply that the way each method is classified is its only use. Rather, the classification suggests that the specified phase for each method is the most appropriate one for that method.

The intent of Figure 6 is to show the suitability of using the authors' analytical framework to compare various IE methods with respect to their coverage. Using the framework, potential users of IE methods can create their own DB-space graph for the IE methods they are using in their organization. The resulting graph would be useful as a basis for comparison among methods used. As a result, the potential users of the IE methods are assisted in: (a) providing a basis for comparison between methods; and (b) the task of selecting appropriate IE methods to cover the entire ISA development life cycle.

Coverage and evolution

In Figure 6, the general shape of coverage and overlap among the methods selected should be noted. This comparison shows that none of the tools, techniques, or methodologies selected can support the entire breadth and depth of the framework. Therefore, a set of methods to cover activities from organizational analysis to systems implementation must be employed.

The heaviest concentration of methods is in the lower-right quadrant. This observation is not surprising because information systems development has been historically oriented towards practical systems development.

On the other hand, the coverage in the upper-left quadrant is light and fragmented. This situation can be characterized as "the search for the holy paradigm." Although there are good ideas causing current excitement over the upper-left quadrant, lacking is a coherent set of IE methods for organizational analysis that provides linkage between organizational strategies and information systems requirements. Figure 7 shows the evolution of IE methods in three stages. The first stage (in the lower-right quadrant) consists strictly of techniques and tools for application development and corresponds to most of the first two generations described in Couger, et al. (1982). The second stage shows: (a) broader systems development techniques (in the middle-right quadrant); and (b) emergence of methodologies for organizational analysis (in the upper-left quadrant), which are disjointed from the techniques in the first stage. The third stage (and current situation) illustrates the emergence of IE methods to link the first two stages.

In summary, although most industry practice is centered on techniques for application development, there are two current trends:

1. Paradigms for strategic planning of information systems, and
2. Computer-aided tools for systems implementation.

As the field matured, industry realized that IE had to move towards a global view of the entire organization's business rather than relying on specific applications. Diverse technology forces a rethinking of fundamentals (e.g., the transition from batch to interactive computing), resulting in requirements for a more solid conceptual foundation.

Lack of organizational analysis focus

Organizational analysis is unfortunately the weakest phase in the breadth dimension (Yadav, 1985). As stated by Zachman (1982), organizational analysis is in its formative stages; however, every business that continues to grow and evolve is likely to employ organizational analysis in some form. Although initial methods (such as Business Systems Planning) were proposed over 15 years ago, the quality and usefulness of the results are largely dependent on the creative abilities of the analysis team, rather than on a well-formulated method.

Methods for organizational analysis ideally should meet three criteria (King, 1985):

1. Relate information systems architecture to the existing organizational strategy, so that a change in organizational strategy would be supported by the information systems architecture.
2. Assess IS resources to identify potentially useful changes in organizational strategy.

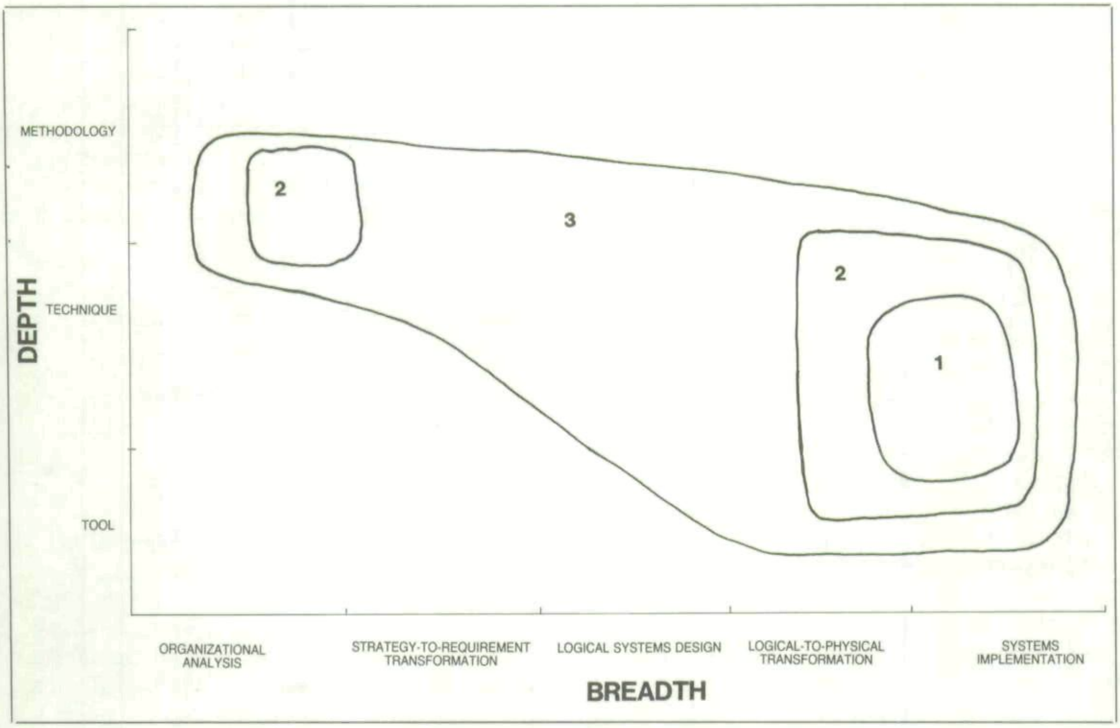

Figure 7. Evolution of Methods

3. Incorporate the notion of IS resources as a strategic resource (or "competitive weapon") and involve ways of identifying opportunities to use those resources.

Therefore, organizational analysis needs to connect, in both directions, strategic planning across the organization with strategic planning of the IS resources. The authors conclude that it is not simply a one-way, top-down activity.

Processes of align and exploit

A dialogue or a process should occur between the conceptual planner and pragmatic developer, as shown in Figure 8. The mission and directions generated by the conceptual planner should flow to the pragmatic developer as the basis on which systems are developed. Likewise, there should be a reverse flow of information from the pragmatic developer to the conceptual planner about the constraints and opportunities of information technology.

These flows are referred to as the processes of *align* and *exploit*, respectively. The align process forces information management to conform to the mission and policies of the organization, while the exploit process searches for opportunities that are feasible given the organization's resources and general state of technology. Thus, a continuous planning process should be established that consists of:

1. Aligning the development and operation of information systems to the strategic plans and directions of the organization.
2. Exploiting the advantages of the existing information technology to change the nature of competition and/or to move into new areas of business.

In this situation where the conceptual planner interacts directly with the pragmatic developer (Figure 8), the differences in conceptual levels (along the depth dimension) cause the exchange between these roles to be ineffective. The conceptual planner states requirements in terms of high-level organizational needs, while the pragmatic developer states capabilities in terms of low-level technical terms. A significant gap in interpreting the respective situations occurs.

As shown in Figure 9, the authors concluded (as did Benson and Parker, 1985) that this process

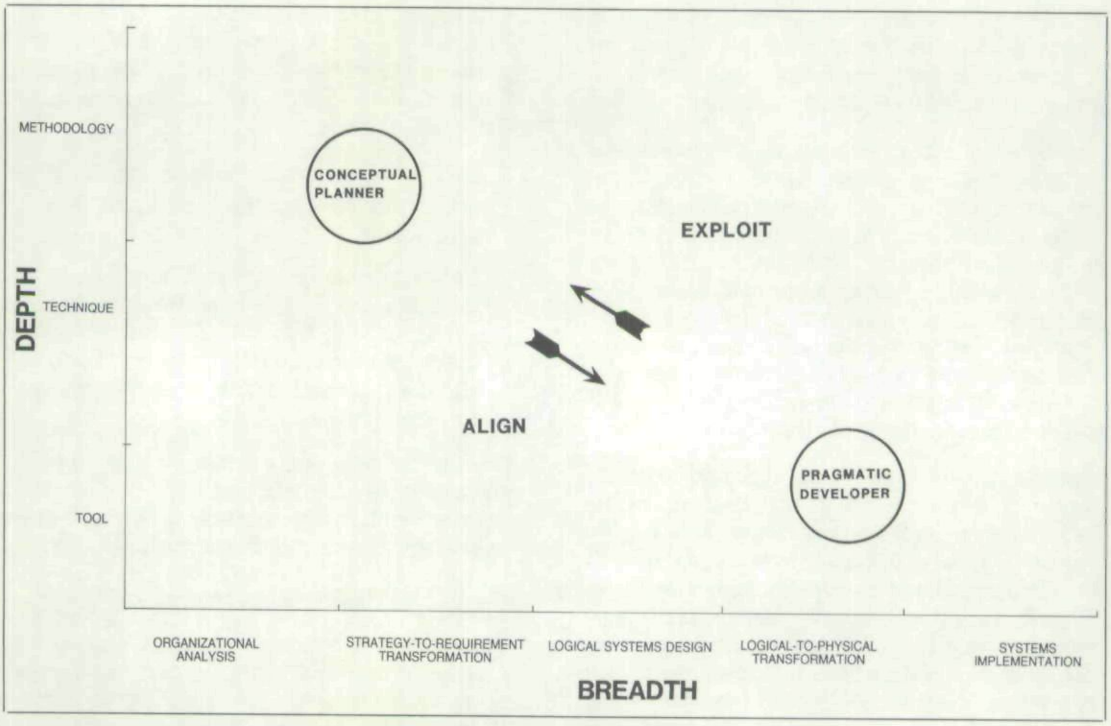

Figure 8. Align and Exploit Between Planner and Developer

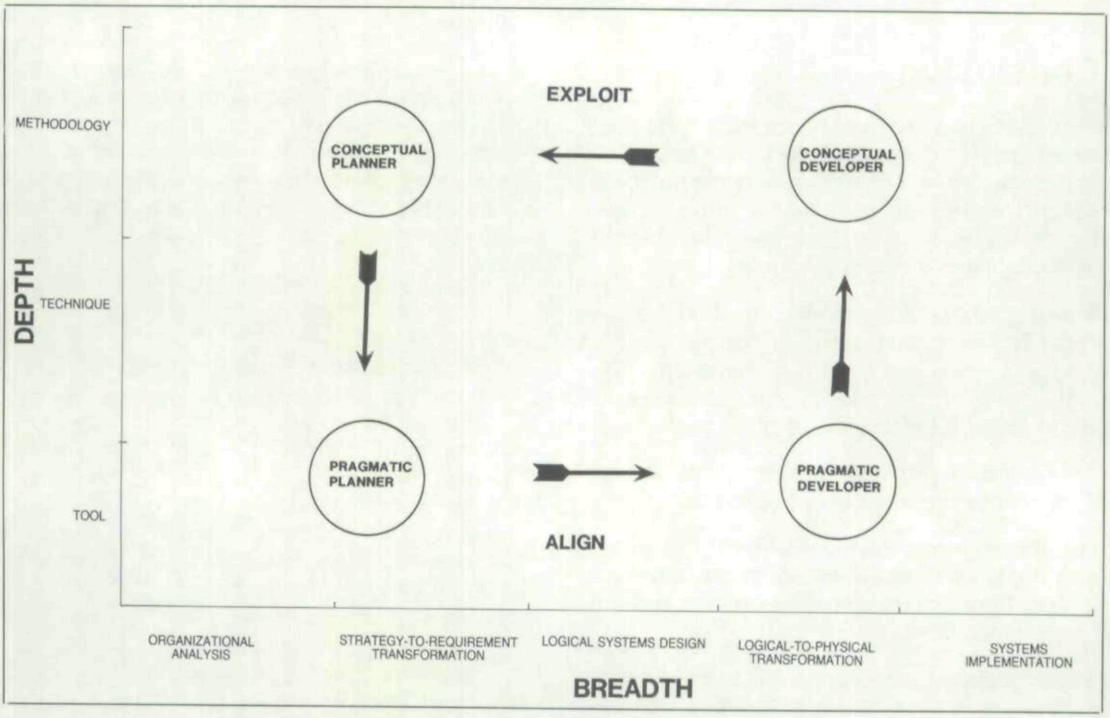

Figure 9. Extended Process for Align and Exploit

should be a counterclockwise flow. Hence, the align process should involve the role of pragmatic planner, and the exploit process should involve the role of conceptual developer.

Establishing additional linkages is essential for an organization in several ways. First, it is important to know how a change in the strategic planning of an organization would affect the planning of information technology because changes in the infrastructure of an organization take years to evolve. It is important to analyze these changes early and identify the cultural issues that are unique to each organization because a significant level of planning and resources is needed to cultural change to happen.

Second, senior executives strongly desire to regain effective control in the aftermath of the information technology explosion. This control can be achieved through strategic planning at both organizational and technological levels in order to align development efforts with strategic business objectives, plans, and priorities. This desire is due largely to senior executives' recognition that information systems are becoming the critical path in effecting critical changes in an organization.

Conclusions

This article has reviewed various methods used for information engineering based on an analytical framework with breadth versus depth dimensions. The findings were illustrated on a graph that indicated a partial coverage currently provided by these methods.

In particular, the IE methods compared are currently limited in their ability to improve productivity of planning and implementation efforts. Several levels of productivity can occur for an organization through the use of IE methods:

1. **Limited** — productivity is limited to the implementation of application projects.
2. **Global** — productivity affects the entire breadth of IE activities within the organization, from organizational analysis to systems implementation.
3. **Strategic** — productivity affects the external responses of the organization to its marketplace, based on its information management.

Industry is currently achieving limited productivity in many situations. However, the authors concluded that other levels of productivity — global and strategic — elude most organizations. The usual reasons are:

1. Lack of IE methods that provide integrated coverage across the align and exploit processes.
2. Lack of a coherent set of IE methods for establishing the linkage between organizational strategies and information systems requirements, and inadequate attention to information management as part of overall organizational planning.

This article provides a coherent framework for comparing IE methods and inferring their management implications, especially as they pertain to strategic planning of information systems.

Acknowledgement

This work was supported through a grant from Coopers & Lybrand, whose support was greatly appreciated. In particular, the insights of Hans Wagner, Bill Inmon, Ed Pringle, and Patricia Foy were valuable. However, the opinions expressed herein are solely those of the authors and do not indicate endorsement by Coopers & Lybrand.

The authors are grateful to Peter Bryant, Feng-Yang Kuo, Jim Gerlach, Doug Durand, and Bill Murray for their comments, along with the constructive comments from the reviewers. The authors also wish to acknowledge the contribution of Debbie Scholz, Kathy Ingraham, and Diane Sewick to this work.

References

AGS Management Systems. "A Family of Methodologies for System Development," King of Prussia, PA, 1985.

Alford, M.W. "SREM at Age of Eight: The Distributed Computing Design Systems," *IEEE Computer* (18:4), April 1985, pp. 36-46.

Appleton, D. "DACOM Approach to Data Resource Management," Version 2, D. Appleton Company Inc., Manhattan Beach, CA, 1985.

Benson, R.J. and Parker, M.M. "Enterprise-wide Information Management: An Introduction to the Concepts," Report Number G320-2768, IBM Scientific Research Center, Los Angeles, 1985.

Blosser, P. "An Automatic System for Application Software Generation and Portability," Ph.D. dissertation, Purdue University, 1975.

Bryan, W.L., Siegel, S.G., and Whiteleather, G.L. "Auditing Throughout the Software Life Cycle: A Primer," *IEEE Computer*, 1982.

Carlson, W.M. "Business Information Analysis and Integration Techniques (BIAIT) — the New Horizon," *Data Base* (10:4), Spring 1979, pp. 3-9.

Chen, P.P. "The Entity-Relationship Model: Toward a Unified View of Data," *ACM Transactions On Data Base Systems* (1:1), 1976.

Codd, E.F. "A Relational Model of Data for Large Shared Data Banks," *Communications of the ACM* (13:6), June 1970, pp. 377-87.

Colter, M.A. "A Comparative Examination of Systems Analysis Techniques," *MIS Quarterly* (8:1), March 1984, pp. 51-66.

Colter, M.A. "Evolution of the Structured Methodologies," in *Advanced System Development/Feasibility Techniques*, J.D. Couger, M.A. Colter, and R.W. Knapp (eds.), John Wiley, New York, 1982, pp. 73-96.

Couger, J.D., Colter, M.A., and Knapp, R.W. (eds.). *Advanced System Development/Feasibility Techniques*, John Wiley, New York, 1982, pp. 6-14.

Database Design Inc. "Data Designer Product Description," Ann Arbor, MI, 1981.

Gane, C. *STRADIS*, Auerbach Publishers, Inc., Boston, MA, 1984.

IBM. "Business Systems Planning," GE 20-0527-1, 1975, pp. 1-92.

Index Technology Corporation. "Excelerator: The Features of Excelerator," Cambridge, MA, 1986.

Inmon, W.H. *Information Systems Architecture*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

Jackson, I.F. *Corporate Information Management*, Prentice-Hall, Englewood Cliffs, NJ, 1986, p. 90.

Karimi, J. "An Automated Software Design Methodology Using CAPO," *Journal of Management Information Systems* (III:3), Winter 1986-87, pp. 71-100.

Karimi, J., Hackathorn, R.D., Scholz, D.L., Ingraham, K.P., and Sewick, D.L. *Evaluation of Methodologies and Tools for Supporting Information Engineering*, Internal Research Report, Information Systems Research Group, College of Business and Administration, University of Colorado at Denver, December 1985.

King, W.R. "Strategic Planning for IS: The State of Practice and Research — Editor's Comment," *MIS Quarterly* (9:2), June 1985, pp. vi-vii.

King, W.R. "Strategic Planning for Management Information Systems," *MIS Quarterly* (2:1), March 1978, pp. 27-37.

Martin, J. *Strategic Data-Planning Methodologies*, Prentice-Hall, Engelwood Cliffs, NJ, 1982.

Mason, R.O. and Mitroff, I.I. *Challenging Strategic Planning Assumptions*, John Wiley and Sons, New York, 1981.

McDonald, C., Riddle, W., and Youngblut, C. "STARS Methodology Area Summary, Volume II: Preliminary Views on the Software Life Cycle and Methodology Selection," prepared for the Office of the Under Secretary of Defense for Research and Engineering, Institute for Defense Analysis. Reprinted in *ACM Software Engineering Notes* (11:2), April 1986, pp. 58-85.

McIntyre, S.C., Konsynski, B.R., and Nunamaker, J.F., Jr. "Automating Planning Environments: Knowledge Integration and Model Scripting," *Journal of Management Information Systems* (11:4), Spring 1986, pp. 49-69.

Mylopoulos, J., Berntsein, P.A., and Wong, H.K. "A Language Facility for Designing Database-Intensive Applications," *ACM Transactions on Database Systems*, Number 5, 1980, pp. 185-207.

Nastec. "CASE 2000 Design Aid Product Description," Southfield, MI, 1985.

Rockart, J.F. "The Changing Role of the Information Systems Executive: A Critical Success Factor Perspective," *Sloan Management Review*, Fall 1982, pp. 3-13.

Ross, D.T. "Applications and Extensions of SADT," *IEEE Computer* (18:4), April 1985, pp. 25-34.

Schouw, R.W. "Life After BSP," IBM Information Systems Management Institute, Los Angeles, 1983.

Teichrow, D. and Hershey, E.A., III. "PSL/PSA: A Computer-Aided Technique for Structure Documentation and Analysis of Information Processing Systems," *IEEE Transactions on Software Engineering* (3:1), January 1977, pp. 42-48.

Wardle, C. "The Evolution of Information System Architecture," *Proceedings of the Fifth International Conference on Information Systems*, Tucson, AZ, November 1984, pp. 205-217.

Warnier, J.D. *Logical Construction of Systems*, Van Nostrand Reinhold, New York, 1981.

Webster New World Dictionary, Third College Edition, William Collins Publishers, Inc., Cleveland, OH, 1981.

Wetherbe, J.C. and Davis, G.B. "Developing a Long-Range Information Architecture," *Proceedings of the National Computer Conference,* AFIPS Press, Anaheim, CA, May 1983, pp. 261-269.

Yadav, S.B. "Classifying an Organization to Identify Its Information Requirements: A Comprehensive Framework," *Journal of Management Information Systems* (2:1), Summer 1985, pp. 39-60.

Yourdon, E. and Constantine, L. *Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design,* Prentice-Hall, Englewood Cliffs, NJ, 1979.

Zachman, J.A. "Business Systems Planning and Business Information Control Study: A Comparison," *IBM Systems Journal* (21:1), 1982, pp. 31-53.

About the Authors

Richard D. Hackathorn is associate professor of information systems at the University of Colorado, Denver. He was a faculty member at the Wharton School, University of Pennsylvania prior to that. He received his B.S. from the California Institute of Technology and his M.S. and Ph.D. from the University of California, Irvine, in 1967. At the end of the spring semester, he will leave the university to start a business in developing education products for the information system field. Dr. Hackathorn is a member of the Association of Computing Machinery, the Society for Information Management, the Decision Science Institute, and the Institute for Management Science.

Jahangir Karimi received his M.S. and Ph.D. degrees in management information systems from the University of Arizona, Tucson, in 1978 and 1983, respectively. Since 1983, he has been with the Department of Information Systems, University of Cincinnati for a year, and the University of Colorado at Denver where he is currently an assistant professor. His research interests include computer aids in the systems development process, software engineering, user interface design, information systems modeling techniques and strategic planning. He has published in the *IEEE Transactions on Software Engineering,* the *Communications of the ACM,* the *Journal of Management Information Systems,* and a number of conference proceedings. Dr. Karimi is a member of the Association of Computing Machinery, Computing Society, and the Society for Information Management.