# Different Perspectives on Information Systems: Problems and Solutions

KALLE LYYTINEN

*Department of Computer Science, University of Jyväskylä, SF-40 100 Finland*

The paper puts information systems (IS) research dealing with IS problems into perspective. IS problems are surveyed and classified. Using the IS research framework suggested by Ives, Hamilton, and Davis, research into IS problems is classified into several perspectives whose relevance in coping with the problems is discussed. Research perspectives focusing on IS operations environment, IS development process, IS development organization, IS development methods, and IS theories are distinguished. The paper concludes with suggestions for future research and how to deal with IS problems in practice.

Categories and Subject Descriptors: D.2.1 **[Software Engineering]:** Requirements/ Specifications; D.2.2 **[Software Engineering]**: Tools and Techniques; D.2.6 **[Software Engineering]**: Programming Environments; D.2.9 **[Software Engineering]**: Management; H.2.1 **[Database Management]**: Logical Design; H.2.7 **[Database Management]**: Database Administration; K.4.3 **[Computers and Society]**: Organizational Impacts; K.6.0 **[Management of Computing and Information Systems]**: General; K.6.1 **[Management of Computing and Information Systems]**: Project and People Management; K.6.3 **Management of Computing and Information Systems]**: Software Management; K.6.4 **[Management of Computing and Information Systems]**: System Management

General Terms: Economics, Human Factors, Management

Additional Key Words and Phrases: Information system failure, information systems development, methodologies, organization of development and use of information systems

## INTRODUCTION

### Background and Motivation

Since the application of computers in administrative information processing began in 1954 [Davis and Olson 1985], computers have become a key instrument in the development of organizations' formal information processing. The rapid development of information technology has helped to firmly establish the general attitude that information systems (ISs) are a powerful instrument for organizational problem solving. This opinion has been strengthened by popular theories of organizational behavior, which view organizations primarily as information-processing systems [March and Simon 1958; Galbraith 1977] or control systems [Landry and Le Moigne 1977; Verrijn-Stuart 1979].

Whereas the above opinion is widely held in the IS community, it is not, however, in line with the evidence gained from studies on how computer technology changes organizational performance [Cron and Sobol 1983; Earl and Hopwood 1980; Ouchi 1978]. In many empirical investigations information systems have been found to be

CONTENTS



rife with difficulties [Kling 1980; Lucas 1975; Turner 1982]. Many researchers have gone even so far as to speak of a "software crisis" or a "systems crisis" [Bubenko 1983; Martin 1985].

Many recognized difficulties with the use and development of IS provide a starting point for our discussion. The goal of this paper is to investigate how the IS community has responded to IS difficulties. Briefly, we put into perspective IS research dealing with potential IS problems, and show that there are a number of ways of perceiving and understanding these problems. We classify major IS problems and explore a number of approaches and perspectives that have been suggested for resolving these problems. Each perspective is reflected in theories, models, and research results. In this way, we evaluate the success of various perspectives in coping with particular IS problems and indicate what sort of empirical support exists for such evaluations. The reader is expected to have a general exposure to the problems and issues of systems design as, for example, covered in standard textbooks on information systems [Davis and Olson 1985].

#### Organization of the Paper

To organize the vast and heterogeneous literature on IS problems, one needs a framework into which past and present research can be organized. During the last decade several frameworks for IS research have been suggested [cf. Ein-Dor and Segev 1978; Ives et al. 1980; Lucas 1975; Mason and Mitroff 1973; Nolan and Wetherbe 1980]. These frameworks deal with different aspects of information systems and have different foci. They thus serve as a basis for formulating alternative research problems, and IS research is classified accordingly.

Because our goal in this paper is neither to compare proposed frameworks nor to propose yet another one, we have chosen to apply the IS research framework proposed by Ives et al. [1980]. There are a number of good reasons for using this framework. First, it is a synthesis of many other frameworks and covers their main elements [Ives et al. 1980]. Second, it is more substantial than others since it also focuses on the development of information systems. By delineating the main components of both the use and development of the information system, this framework helps us explore and understand multiple features of the IS and their environments, as well as IS problems and their potential solutions. Third, it is widely known and has been used in the classification of IS research literature [Ives et al. 1980]. Fourth, we found that Ives and his co-workers' IS research framework

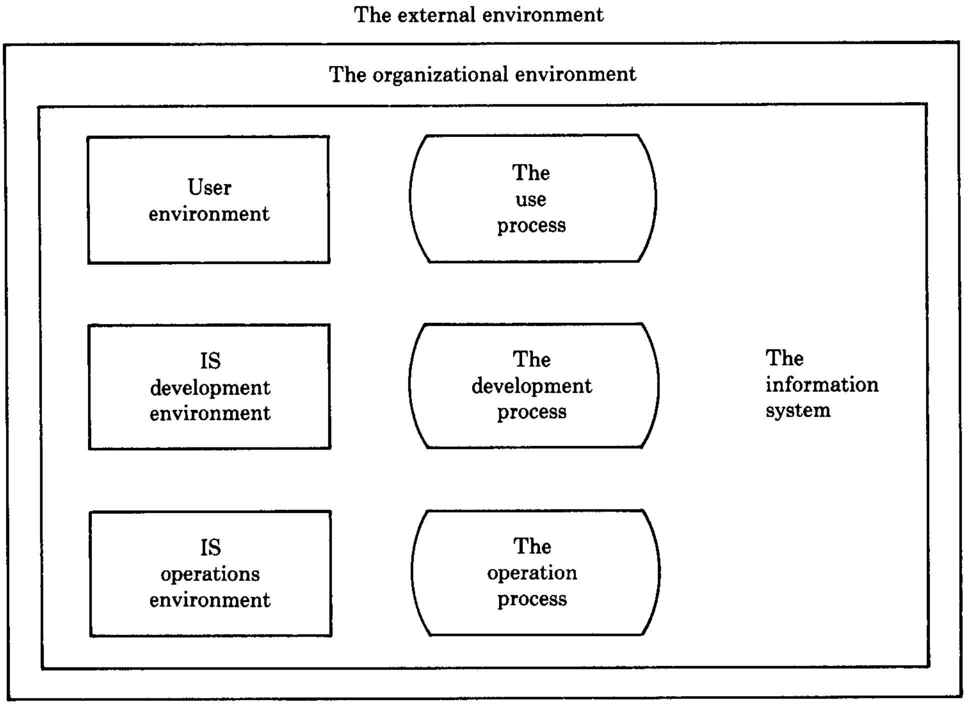


**Figure 1.** A model for IS research in Ives et al. [1980].

proved to be valuable in understanding IS failures and classifying different approaches toward solving them. Thus its credibility is established for understanding more practical aspects of IS.

Since the Ives et al. [1980] model of IS research is widely known and discussed, we only briefly outline its basic tenets here. A pictorial model of the framework showing its main components and their interactions is presented in Figure 1. The main contents of each model component are presented in Table 1. Hereafter we use the terms "IS framework" or "IS context" for this model.

The model distinguishes among three information system environments—user, IS development, and IS operations environments (represented by squares)—and three information system processes—use, development, and operations processes (represented by ellipses). The environments define the resources and constraints that dictate the scope and form of information systems and IS processes. The IS processes comprise dynamic interactions among the IS, IS environments, and other IS processes. The information system, its environments, and IS processes are surrounded by two more extensive environments: the organizational and external environments. All these elements are called IS context components.

In this paper this model is used to identify and organize the vast literature dealing with IS problems in two ways. First, in Section 1 we classify the main IS problems by identifying the major deficiencies of the three IS processes. Second, in Sections 2–6 we use the framework to locate perspectives to study and solve IS problems. By perspective we mean a standpoint that selects specific components and their interactions in the IS context for the study. A perspective also incorporates assumptions and conjectures regarding each component's behavior and how the selected field of study can be investigated. In each section we focus on one perspective by exploring its main research results and analyzing how they address different IS problems. Each

**Table 1.** Components of the IS model

| Component | Content |
|---|---|
| The information system | The system that processes information; characterized by content, form, and time of presentation |
| IS operations environment | Resources necessary for IS operations; components include software, hardware database, procedures/documentation, organization, and management of IS operations |
| IS development environment | Development methods and techniques and their characteristics, design personnel and their characteristics, and the organization and management of IS development and maintenance |
| The user environment | Primary users of the IS such as decision makers and intermediaries |
| The organizational environment | Organizational goals, tasks structure, volatility, and management style and culture |
| The external environment | Legal, social, political, cultural, economic, resource educational, and industry/trade considerations |
| The use process | Use of the IS by the primary user |
| The development process | The selection and application of organizational resources that yields the IS |
| The operations process | The physical operation of the IS, which is primarily a function of the operations resource |

section concludes with a summary of how the perspective can help resolve specific IS problems, noting empirical studies that demonstrate this.

The five perspectives analyzed are derived from the components inside the inner rectangle in Figure 1 because a fundamental idea in the mainstream of IS research has been that improvements in the three IS processes result from changes in the three immediate IS environments, the organizational environment, and associated IS processes. The impact of changes in the external environment has been summarized in more detail in some recent surveys [cf. Hirschheim 1986; Klein and Hirschheim 1987; Kling 1980], which are excluded from our field of study.

In Section 7 we arrive at several conclusions concerning research needed in future and how to apply suggested measures in practice.

Because the IS field, as delineated above, covers a complex and large field of inquiry, we have combined results from several disciplines, and the result is a profile from several thousand references dealing with the development and use of information systems. The literature referenced is not exhaustive, but an attempt is made to cover widely known works or original research results. In addition, we have tried to include references to both North American and European research. Some relevant areas, such as software engineering economics [Boehm 1981] and human factors engineering [Ledgard et al., 1981; Moran 1981; Shneiderman 1981], have not been covered in depth.

## 1. PROBLEMS WITH THE INFORMATION SYSTEM PROCESSES

An exact characterization of IS problems is difficult—perhaps impossible—to make. As noted by many [Lucas 1975; Morgan and Soden 1973; Turner 1982], an IS "failure" is multidimensional and has several subcomponents: technical, behavioral, political, etc.

In the following we discuss IS problems in two IS processes: development and use. This discussion is based on a more detailed survey by Lyytinen and Hirschheim [1987].

### 1.1 Problems with the Information System Development Process

The difficulties with information systems development have been widely discussed in books, seminars, and workshops for over

**Table 2.** IS Development Process Problems

| IS problem class | Description | References |
|---|---|---|
| Goals | Goals are ambiguous, too narrow, and conflicting | Senn [1978], Kumar and Welke [1984], Keen and Scott Morton [1978] |
| Technology | Technology restricts choices, high risk of change | Alter [1980], Kling and Scacchi [1982] |
| Economy | Poor quality of calculations; lack of foundations | Mayntz [1984], Kling [1980], Hirschheim and Smithson [1987] |
| Process features | Analysts dominate; poor communication; lack of quality control | Senn [1978], DeBrabander and Thiers [1984], Bubenko [1986] |
| View of organization | Neglect of behavioral and organizational issues | Hedberg [1980], Keen and Scott Morton [1978], Mumford [1981] |
| Self-image | Highly rationalistic image | Kling [1980], Kling and Scacchi [1982], Kumar and Welke [1984] |

two decades [Brooks 1974; Lucas 1975, 1981]. There are many indications that IS development is fraught with recurrent problems caused by poor, undisciplined, and incomplete development practices. According to a recent survey referenced by Gladden [1982], 75 percent of all systems development undertaken was never completed or, if completed, not used. In a similar vein, Canning [1977] points out that the inordinate amount of total life-cycle costs (70 percent) spent on systems "maintenance" is a symptom of poor development practice, and that the development process, especially its early phases, is of low quality [Bell and Thayer 1976].

On the basis of a classification by Thoresen [1984], we divide IS development process problems into six classes: goals, technology, economy, process features, view of the organizational environment, and self-image. The main content of each problem class is shown in Table 2.

### 1.2 Problems with the Information System Use Process[1]

If information systems were truly "problem solvers," using them would improve decision making and increase organizational effectiveness. Unfortunately, empirical studies provide some evidence suggesting that this is not often the case. An extensive assessment of users' opinions about the IS use process can be found in Lucas [1975], who portrays the deficiencies in the IS use processes as follows:

> ... users do not understand much of the output they receive; there is duplication of input and output and changes are frequently made in systems without consulting users. Because of inaccuracies, users often discount all information provided by the system. Many users complain of information overload: massive amounts of data are provided which cannot be digested by the decision-maker.... A number of users report that they do not actually use the information provided by the system. Many feel that computer-based information systems are not worth the time or cost to develop and that organizations would be better off without them. [Lucas 1975, pp. 2–3]

Similar conclusions have also been reached by Pearson [1977], Lucas [1981], and Turner [1982]. Although more positive user attitudes toward information systems have been reported [Adams 1975; Senn 1980], the general picture is pessimistic.

Thus the IS use process encompasses a rich variety of problems. Following Alter [1980] we divide these into five problem categories: technical, data, conceptual, people, and complexity. A description of the main contents of each problem class is shown in Table 3.

[1] The IS use process and the IS operations process are discussed together, and they are not conceptually distinguished. As the operations process is instrumental for the use process, this unification is natural. Moreover, the literature seldom distinguishes between these two processes.

**Table 3.** The IS Use Process Problems

| IS problem class | Description | References |
|---|---|---|
| IS operations problems | IS is difficult to use; interface is awkward; IS is slow and unreliable | Alter [1980], Shneiderman [1981], SEN [1983] |
| Data problems | Data are incorrect; lacks relevance; is incomprehensible or missing | Alter [1980], Nygaard and Håndlykken [1981], Boland [1979], Senn [1979] |
| Conceptual problems | Wrong problem solved | Mitroff [1980], Alter [1980] |
| People problems | Negative impact on work, power shifts and job qualification changes | Attewell and Rule [1984], Kling [1980], Hirschheim [1986], Göranzön [1983] |
| Complexity problems | IS is too complex to understand, maintain, and use | Rosenberg [1980], Buckingham [1977], SEN [1983] |

### 1.3 Summary

Empirical studies of information systems show that IS processes are fraught with several types of difficulties, which are rich and varied and come from many backgrounds. Particularly, the quality of the IS development process is often poor, which also detracts from the quality of the IS use process.

We can calculate that the cost of all IS problems in the United States is several hundred million dollars annually. When we add to this the observation that empirical data about positive intangible computer impact are largely ambiguous [Attewell and Rule 1984; Hirschheim 1986], we can conclude that the situation poses a continuing challenge to the IS community. There is a great practical need to understand the extent, impact, and nature of IS problems, as well as the proficiency of various approaches toward resolving them.

The classification of IS problems presented here requires some additional comments. First, we have not explored the relationships among different IS problems, even though IS problems are strongly interdependent. For example, many IS development process problems correlate highly with several IS use process problems. We have not, however, analyzed these relationships because there are hardly any empirical studies on the topic, a full account of which is beyond the scope of this paper.

Second, we have not described in detail how common various IS problems are. The knowledge of, say, the number of information systems in which we have data or conceptual problems would help in assessing the importance of improvements in the IS context components. Unfortunately, no extensive empirical studies have been done on this subject. Several researchers have found [Alter 1980; Lucas 1975] that most IS problems relate to the social and conceptual aspects of the IS, a finding that was ascertained in a recent empirical study [Lyytinen 1987b]. In the IS development process the three most common problems involve goals, process features, and self-image. In the IS use process the three most common problems involve complexity, concept, and people's reactions.

## 2. TECHNICAL ADVANCES IN OPERATIONS AND DEVELOPMENT ENVIRONMENT

Research of IS in this area suggests that technological breakthroughs in the operations and development environments will remove IS problems. Thus new technology opens several avenues to improving IS processes. Also, technological changes in computing environments will drastically change the IS operations process. On the other hand, widespread use of capital-intensive software technologies will change the IS development environment and improve the development process.

### 2.1 Changing Computing Environment

The current state of the art in computer technology is characterized by so-called fourth-generation computing systems [Canning 1983, 1984; Gupta 1982]. These distributed systems offer user-friendly interfaces, higher reliability, and better performance/cost ratio than their predecessors, and they are substantially changing the form and organization of the IS operations process.

Fourth-generation technology, however, is not a safe means of combating IS problems. First, new technology affects the operations process on personal or group levels. What its impact will be on organization-wide information systems is far from clear, and many IS problems concern such organizational impacts. Second, every computer generation necessitates new approaches toward understanding the IS use process and also toward establishing it on a wide organizational basis. These approaches are just beginning to emerge, and their adoption into practice will take time.

### 2.2 Capital-Intensive Software Technologies

A production process is capital intensive if it involves large early expenditures to increase the overall productivity of the process [Wegner 1982], and information systems development is becoming highly capital intensive. This happens in two ways. First, systems design efforts are extensively channeled through computer-based support environments. Second, software development is very often carried out with application generators, which are extremely capital intensive.

#### 2.2.1 Development Support Environments

Development support environments integrate sets of tools that an IS developer can use to accomplish specific development tasks [Wasserman 1980, 1982; Waters 1974]. They usually form an integral part of the operations environment in the same way as editors and compilers. Examples of such environments are USE [Wasserman 1980, 1982], PSL/PSA [Teichroew et al. 1980], and PLEXSYS [Nunamaker and Konnsynski 1981].

Support environments ease the tedious maintenance of development documents and help to derive complete specifications, check consistency [Welke 1981], and manage the complexity with modular designs. In addition, they shorten development times by generating program codes and helping to predict the performance of the system.

Experience with these environments shows, however, that they are mainly helpful in developing large information systems and are thus cost effective in organizations with a large IS function. In addition to considerable investments in software, they require extensive education in their use and the availability of sufficient computing power. Another shortcoming originates from their limited scope. Most support environments are targeted to the system's design tasks and neglect many other aspects of the IS design: organization development, requirements analysis, and so on.

#### 2.2.2 Application Generators

Application generators exploit common features of information systems by providing a set of generic functions [Horowitz et al. 1985]. They provide a high degree of functional abstraction compared with traditional high-level languages, and thus can improve development productivity. Information systems can often also be developed by the users, as witnessed by the rapid increase in end-user computing (EUC) [Martin 1982, 1985].

Thus application generators seem to prevent many IS problems from occurring. Unfortunately, this applies only to some of the problems. For example, they do not remove IS problems related to goals. Moreover, they are limited in their function in that they cannot be applied to the development of all types of information systems. In applications that involve, for example, heavy computational loads like the bill of materials explosion, their limitations can become an obstacle. The acquisition and maintenance of application generators may also create new technical and operational problems.

### 2.3 Summary

A number of researchers and industry analysts believe that advances in technology will solve many of the major IS problems (see, e.g., Martin [1985]). This, however, is

**Table 4.** IS Problems and the Impact of Technical Advances

(a) IS Development Process Problems

| Changes | Goals | Technology | Economy | Process | View of organization | Self-image |
|---|---|---|---|---|---|---|
| Computing environment | — | R | P | P | P | — |
| Support environment | — | — | R | P | — | — |
| Application generator | — | — | R | P | P | — |

(b) IS Use Process Problems

| Change | Operations | Data | Conceptual | People | Complexity |
|---|---|---|---|---|---|
| Computing environment | R | P | — | P | P |
| Support environment | R | P | — | — | R |
| Application generator | R | P | P | P | P |

not necessarily the case. A better IS operations process does not necessarily equate with a better IS use process [Markus 1983], as the IS use process is not merely a matter of technical operation [Courbon and Bourgeois 1980].

In spite of all this, our study does show that advancing technology offers great *opportunities* for improving all IS processes, and that it will play an important role in shaping new types of information systems [Dickson et al. 1984].

Table 4 is a summary of the ways in which IS problems relate to technical advances. Three types of connections are distinguished for each problem class and technology improvement: relevant (R), partly relevant (P), and irrelevant (—). Relevant means that we have some empirically demonstrated evidence that the technology change can improve solution potential for the IS problem class in question. For example, many studies show that development support environments can substantially improve the technical quality of the information system. Partly relevant means that empirical evidence of how the IS context component change affects the problem area is ambiguous. For example, on several occasions, application generators can reduce data problems because they promote user involvement and direct prototyping, and focus on the contents of data. This does not imply, however, that data problems have been solved or reduced just by using application generators, or that application generators are always instrumental in this task. Irrelevant means that we could not find empirically demonstrated evidence that the change has any direct impact on the problem area. For example, the type of computer technology seems to have no impact on what the developers think of the nature of the development process, that is, its self-image.

The problem–component change matrices shown in this paper are largely based on our critical assessment of the literature, experience, and many discussions with colleagues, and in this sense they are not based on strict empirical data. We first tried to find empirical support that would corroborate or falsify the claims presented. Unfortunately, we quickly observed that there is hardly any literature that tests the empirical validity of the types of connections shown. Therefore, a set of initial problem–change matrices were discussed with several IS researchers to test for their validity and correctness. As a result, several amendments and deletions were made, which resulted in the matrices shown in this paper. Thus the matrices can be best read as a set of empirical research conjectures.

## 3. DEVELOPMENT PROCESS CHANGES

This body of IS literature suggests that IS problems can be removed by improving the IS development process structure. This structure is normally represented by a *process model*, which provides an interpretation of the IS development process by enumerating, classifying, specifying, and sequencing development tasks. It is a normative model—an idealization of how the IS development process ought to proceed. A typical example of a process

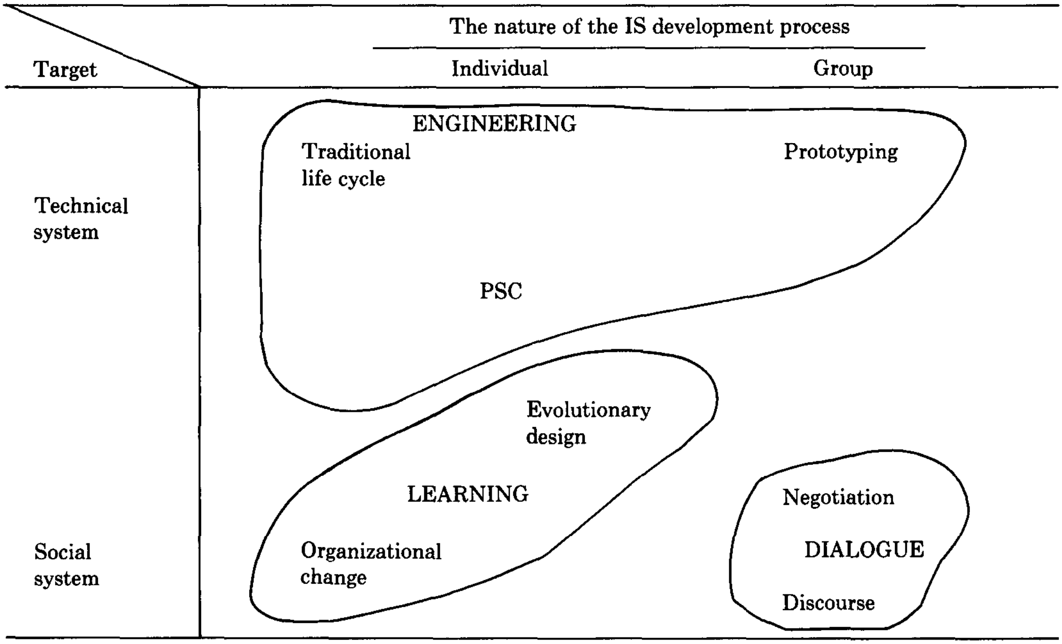


**Figure 2.** Process model types.

model is a "waterfall" systems life-cycle model, suggested by Boehm [1976] and many others for the software engineering process.

A process model is subsumed under a *process model type*; that is, a process model type is a higher level abstraction of a set of process models. To keep things simple, we suggest that process models can be classified into process model types according to two independent dimensions: the target of the IS development process and the nature of the IS development process. These dimensions form continua along which the process models can vary.

The development target concerns either the information system or the larger surrounding social system (user and organizational environments). The nature of the IS development process dimension determines whether the development process concerns either an individual or a group. In the former case it is an individual problem-solving process. In the latter it is a collective social coordinative process. Using these dimensions, we distinguish the following seven process model types (Figure 2): life cycle, prototyping, PSC model, evolutionary design, organizational change, and discourse process.

We further classify the seven process model types into three process model type classes: *engineering, learning,* and *dialogue.* This grouping leads to distinguishing three regions in the figure, where the main characteristics of the IS development process are engineering, learning, and dialogue, respectively. *Engineering process model type class* suggests that the IS development process mainly involves the engineering of technical (and social) artifacts. *Learning process model type class* views the IS development process as individual and collective learning. *Dialogue process model type class* sees the IS development process as a collective inquiry and bargaining process that requires a mutual dialogue.

### 3.1 Engineering Process Model Type Class

#### *3.1.1 Life-Cycle Model Type*

Conventionally the IS development process has been founded on the idea of a product life cycle [Hammer 1981]. Information systems development consists of activities that comprise the software development and evolution of the system [Wasserman and Freeman 1983]. A life-cycle concept is further characterized by concepts of functional design, an individual's rational behavior, and representation by refinement and transformation.

The life-cycle process model type applies a functional strategy [Lanzara 1983; Yordon 1982]. The design outcome is iteratively decomposed into constituent parts and then recomposed into its final form [Langefors 1973]. The decomposition is done on consecutive levels of abstraction, which range from application concepts to concepts applicable to a description of the operational system. On each level the system's functions are described in greater detail. Thus every component in the final design has a function that is precisely described, and there exists a step in the design process that specifies and implements this function. Ideally, we have an isomorphism between the morphology of the design process and the morphology of the information system.

In the life-cycle model the design proceeds by an individual's actions. A designer exhibits completely rational behavior in the sense of a rational view of decision making [Keen and Scott Morton 1978]: He or she has complete control over design options and can predict the outcomes of every design decision [Lanzara 1983].

From a representation viewpoint, a design process is a sequence of linguistic transformations [Lehman 1984], which involves transformations that map the problem statement into the problem solution proposal. In this sequence each representation involves the idea of both the problem and the system to be designed.

The strength of the traditional life-cycle model type is its emphasis on the predictability, stability, and control of the development process and its account of rational problem-solving behavior. Its shortcomings are inflexibility, the limited scope of the target, and neglect of the ambiguous context in which any design process is carried out [Lanzara 1983; Lehman 1980; Parnas and Clements 1985].

#### 3.1.2 Prototyping Process Model Type

Within the engineering process model type class the life-cycle model type has been recently subjected to growing criticism: It is pointed out that the idea of a sequential task flow is overly idealistic and does not correspond to real-life experience. In addition, critics challenge its implicit assumption that requirements can be stated in advance once and for all by a "specification freeze." The life-cycle concept further neglects the learning involved in the experimental design and production of many technical artifacts. As prototypes are used with success in most areas of mass production, why cannot this strategy be used in the IS design? Therefore a prototyping process model type which suggests that its requirements and functions are developed in parallel with their implementation has been recommended [Appleton 1983; Jenkins 1983; McCracken and Jackson 1981]. This incorporates group experimentation and learning into the early phases of the IS development process.

In the prototyping process model type the IS development process involves several transformation sequences, not just one. These can be achieved by applying appropriate operations and development environments: application generators and support environments.

The prototyping approach has an engineering orientation because it limits the scope of the design to a technical system. The concept of the life cycle is only partly abandoned and will be followed when the prototyping phase has found out the correct functions for the information system.

#### 3.1.3 PSC Model Type

Another revision of the life-cycle model type is the Finnish PSC systems development model [Iivari 1983; Kerola 1980; Kerola and Freeman 1981]. The PSC model has been developed from the classical life-cycle model, but extends and transcends it in many ways insofar as it recognizes the need for a simultaneous treatment of all abstraction viewpoints during each developmental step. In the conventional life-cycle model type this is accounted for only by implicit iteration loops. Parallelism is achieved by dividing the development process into three hierarchic and sequentially ordered control and decision-making levels called the pragmatic (P), the semantic (S), and the constructive (C) levels; hence the name PSC model. On the pragmatic level one studies the impacts and purpose of the system, on the semantic level one studies the principal functional input–output relations in the system, and on the constructive level one designs and implements the system.

When comparing the PSC model with the life-cycle model type, the inclusion of P-level tasks in the system's development process is a powerful enhancement. The PSC model has a larger target than the life-cycle model type and thus incorporates the learning component into the IS development process [Iivari 1983]. Also, iteration and recursion become visible, making the PSC model type more adaptable than the life-cycle model. For example, prototyping can be easily accommodated into the PSC model type [Iivari 1983]. The model also recasts the idea of completely rational behavior applied in the life-cycle model type and replaces complete rationality with a bounded one [Newell and Simon 1972] (see Klein and Welke [1980]).

On the other hand, the PSC model retains engineering orientation: It frames the development process with rational decision-making metaphors [Iivari 1983]. Also, owing to its complexity, it is difficult to apply and operationalize [Klein and Welke 1980].

### 3.2 Learning Process Model Type Class

#### 3.2.1 *Evolutionary Design Model Type*

"Evolutionary" or "middle-out" development strategies [Lucas 1978; Keen 1981a] challenge the structuring of the development process in terms of the hierarchic decomposition of activities. They emphasize the role of organizational learning in effective IS development processes: A successful IS development process is more a matter of social learning. The information system is an incremental outgrowth of this learning, and it continues to evolve over time owing to new learning experiences.

Here the IS development process takes place through a user's learning cycle. Therefore we can speak of immature and mature phases of the IS development process [Lucas 1978] that correspond to the user's different learning stages.

Evolutionary design has been applied primarily to single-user or group-based information systems, with a strong orientation toward decision support. Usually these systems lack a clear statement of requirements. In this sense, evolutionary designs are confined to small-scale information systems, and their applicability to other areas remains to be demonstrated [Keen and Scott Morton 1978].

#### 3.2.2 *Organizational Change Process Model Type*

In the organizational change process model type, the idea is to view the IS development process in terms of organizational change strategies that the developers can effectively use to improve chances of successful IS implementation. Here the focus is primarily on changes in the user and organizational environments. Two theoretical perspectives have been suggested [Ginzberg 1978]: *planned change models* and *innovation models.*

Planned change approaches originate from the Lewin–Schein [Lewin 1952] planned change model, in which an organizational change involves three stages: unfreezing, moving, and refreezing. Kolb and Frohman [1970] have refined this further by suggesting a sequence of seven steps: scouting, entry, diagnosis, planning, action, evaluation, and termination. Each step is concerned with the changes in the balance of forces existing in the organization and the degree to which they foster IS change or resistance to it.

The planned change approach has been used in decision support system implementations [Alter 1980; Alter and Ginzberg 1978; Keen and Scott Morton 1978], and its validity has been tested in several studies [Ginzberg 1978; Sorensen and Zand 1975].

An innovation process approach delineates a sequence of steps followed during the adoption of an innovation. This model has also been applied and its validity tested [Ginzberg 1978]. The innovation process is reflected in some development methodologies. For example, sociotechnical approaches [Mumford 1983] and the ISAC methodology [Lundeberg et al. 1981] emphasize the importance of recognizing the problem opportunity and developing an awareness of potential solutions.

The strength of the organizational change model type is its focus on the motivation for learning new organizational behaviors. In this sense, it goes beyond the evolutionary design model type, which neglects processes that can foster organizational learning. The weakness of the organizational change model type is that it focuses on studying an individual's behavior, which can effect or prevent the change. Consequently, the change is not seen as collectively created, but as depending more on the individual's attitude.

### 3.3 Dialogue Process Model Type Class

#### *3.3.1 Bargaining Model Type*

In the organizational change model type, an organization is viewed as a self-regulating system that adapts to its environment by learning. In contrast, in the bargaining model type, an organization is seen as a negotiated order, and the IS development process is a political intervention into a negotiated status quo [Kubicek 1983]. As a result, the IS development process is a mixed political conflict–cooperation game. Through an IS development process, organizational groups can gain or lose power as a result of their bargaining position and skill. The groups improve their strategic position through internal cohesion and education, and political coalitions.

There are several variations of the bargaining model type. The most prominent example is the "systems development by negotiations" approach used in several Scandinavian countries, especially Norway [Ciborra and Bracchi 1983; Kubicek 1983; Nygaard 1983; Schneider and Ciborra 1983] because in Norway a labor–management agreement is needed to develop any computer-based system. These technology agreements have established a variety of rights for workers when bargaining over the design of information systems. From the union perspective, "systems development by negotiations" allows workers to obtain new knowledge, decrease the risk of being overpowered, create solidarity, and participate actively [DEMOS 1979]. Other types of political process models have been discussed by Keen [1981b], Hirschheim et al. [1984], and Robey [1984].

The strength of the bargaining model type is that it recognizes conflict in the IS development process and that this conflict has to be handled by bargaining and political maneuvering [Robey and Markus 1984]. The weakness of the bargaining model type is that it does not say much about what conflict resolution strategies are available and the conditions for effective conflict resolution.

#### *3.3.2 Discourse Model Type*

In a discourse model type an IS development process is a joint search for finding rational reasons to act, which takes place through an exchange of arguments, that is, through discourse. The discourse is aimed at the sharing of language, mutual understanding, and trust. The exchange of arguments exhibits diversified pictures of organizational reality that underlie suggested acts [Lanzara and Mathiassen 1985]. A discourse must bring different and extreme points of view together in the task of generating and interpreting IS development tasks. This requires democratic dialogue that critically searches for IS problems and their solutions.

In a discourse model type there is a no "life cycle" of the output. Instead, the IS development and use processes form a continuous symbolic process involving the creation, sharing, maintenance, and criticism of organizational pictures. These pictures

**Table 5.** IS Problems and the Impact of Development Process Changes

(a) IS Development Process Problems

| Change | Goals | Technology | Economy | Process | View of organization | Self-image |
|---|---|---|---|---|---|---|
| Life cycle | — | R | P | P | — | — |
| Prototyping | P | P | P | R | P | — |
| PSC | P | P | R | R | P | — |
| Evolutionary approaches | P | — | P | R | P | — |
| Organizational change | P | — | P | R | P | — |
| Bargaining | R | — | P | R | P | R |
| Discourse | R | — | — | R | P | R |

(b) IS Use Process Problems

| Change | Operations | Data | Conceptual | People | Complexity |
|---|---|---|---|---|---|
| Life cycle | R | P | — | — | R |
| Prototyping | R | P | P | P | P |
| PSC | R | P | R | P | R |
| Evolutionary approaches | — | P | R | P | — |
| Organizational change | — | — | P | R | — |
| Bargaining | — | P | R | R | — |
| Discourse | — | R | R | R | — |

play an active role in defining problems, proposing agendas, and providing exemplars for organizational action. Examples of discourse models are Checkland's soft-systems methodology [Checkland 1981, 1984], Lanzara's view of the development process as "games" [Lanzara 1983], and Boland's [1978, 1981] and Boland and Day's [1982] work on symbolic dimensions of the IS design.

The problem with the dialogue process model type is that it tends to be complex and requires considerable skill and knowledge to apply. There are few proposals as to how to structure the argumentation process. Without an appropriate agenda for debate, however, the IS development process turns into meaningless chatter. One promising approach is Mason and Mitroff's [1981] method for challenging design assumptions.

### 3.4 Summary

In a survey by Dickson et al. [1984] an understanding of the IS development process ranked sixth in importance for IS success. This explains IS researchers' great interest in structuring the IS development process. Other reasons are rapid changes in IS environments and the new types of information systems developed. This necessitates a fresh look at the development process.

The seven process model types solicited exhibit a large variety of views for framing the IS development process. Each focuses on a different component in the IS context for interpreting the development process. For the life-cycle model type the focal point is the technical system; for the organizational change model type it is the user environment; for the bargaining model type it is the organizational climate and political environment of the information system. In this sense, various development process improvements apply to different sets of IS problems. This is shown in Table 5.

As Table 5 demonstrates, a flexible mixture of various process structures is essential for solving all IS problems. This currently lacks a cogent framework. More research is needed to understand the requirements of each component in the IS context sets for appropriately framing the IS development process. Immediate results can be obtained, however, by focusing on the critical problems that each process model type can effectively handle. Life-cycle models deal effectively with technical and operational problems; a prototyping model can be used when developers are uncertain about the functions of the IS; organizational change models apply to development processes where the biggest problems relate to the user's motivation; finally, bargaining and discourse models apply when there is a great uncertainty about the goals and problems to be solved, and the user and political environments are characterized by strong, conflicting coalitions.

## 4. DEVELOPMENT ORGANIZATION CHANGES

This body of IS literature assumes that the success of the information system depends on the social structures and interactions that prevail during and after the development process. Information systems create problems because prevailing social arrangements inhibit creative planning and successful change. The issue of social arrangements for supporting change and creative planning arises at two levels: management of the organizations' IS function and project management.

### 4.1 Management of Information System Function

The management of IS function is concerned with global IS policy. This policy sets guidelines and principles for modeling, managing, and developing the applications portfolio, development and implementation of a technological infrastructure, management and organizational location of the IS function, IS personnel planning and education, methodology selection and education, standards, documentation, information technology forecasting and planning, etc. [Davis 1984].

The global IS policy setting on the organizational level has only recently gained more research attention [cf. Davis 1984; Dickson et al. 1984; Nolan 1984]. There are few, if any systematic studies on its impact on IS problems; the most researched area is probably the modeling of the applications portfolio, which is discussed in Section 5.1. Therefore it is difficult at this stage to evaluate the impact of global IS policy on potential IS problems.

In this section we briefly touch on one aspect of the IS management: centralization versus decentralization of IS function. This issue has been a source of great controversy and heated debate for at least one decade [Emery 1977; King 1983]. Today's general opinion is that each position in the centralized/decentralized continuum has its pros and cons. Therefore, how instrumental decentralization is in solving potential IS problems is currently an open question, and the outcome of decentralization seems to depend on types of computing, management, and user expectations, that is, on the IS type and its user and organizational environments [King 1983].

### 4.2 Project Management

The level of project management deals with managing the development of a particular information system. Project management is possible without a global IS function, although this rarely occurs. Since the early 1970s [Blumenthal 1969] it has been common policy to align specific IS development processes with a more global IS strategy.

There are three problems in structuring a project management strategy, which involve

(1) administrative models,
(2) interaction structures,
(3) interaction strategies.

These issues are also relevant to the management of a global IS function, although emphasis in the current IS literature lies at the level of project management. A common view also seems to be that there are no principal differences as to how the three issues are handled at the two levels.

The three issues are hierarchically related so that administrative models define specific interaction structures, which, in turn, mold the interaction strategies. Therefore, the same order is followed in the following discussion.

#### *4.2.1 Administrative Models*

Administrative models are concerned with the formal mechanisms that enable participant groups to take part in the development process. These mechanisms take various forms, depending, for example, on the process model type and development methods followed.

The traditional administrative model consists of a decision-making hierarchy that allocates resources, defines performance standards, and audits systems development results. The model sets rigid reporting and control structures by defining roles that are responsible for well-defined tasks [Ciborra and Bracchi 1983; Scacchi 1984]. The development organization is staffed mainly by IS professionals. The users exert their influence by having their representatives participate in the system's administrative machinery. Their role is to review and accept the specifications and to give information needed by IS professionals.

In recent literature new administrative models have been suggested, including matrix-type project organizations, parallel project organizations [Lundeberg et al. 1981], cooperative strategies, and user-controlled systems development. Mumford [1983] distinguishes between vertical and horizontal administrative models. The former involves user's representation on all hierarchic levels of the project organization, and the latter specifies a wide organization that allows all affected to participate directly. More attention has also been paid to "contingencies," such as development contracts and their content [Beath 1983] or social externalities [Ciborra 1983] that will affect the selection of an appropriate administrative model. For example, Beath [1983] identifies clans, bureaucracies, and markets as different organizational archetypes that can be applied, depending on the content of the development contract. The choice depends on the uncertainty and chance for opportunism associated with contracting.

The type and number of roles identified by an administrative model have also undergone a rapid change. Many scholars have challenged the classical dichotomy between IS professionals as primary problem solvers and users as passive clients. Boland [1978] suggests the problem-solver role for all IS developers. The primary difference between IS developers is the specific perspective in viewing the problem—not problem-solving ability. Similar arguments have also been put forward by Lundeberg et al. [1981] and Mumford [1983]. Mumford points out that the user as a problem solver also shapes the role of the IS professional in a new way—from that of "designer as sole problem solver" to that of "designer as teacher, advisor, and agent of change." In the same vein, Keen and Scott Morton [1978] argue in favor of the changing role of IS professionals as implementors and agents of change.

#### 4.2.2 Interaction Structures

An interaction structure suggests a pattern of communication between different administrative roles. It includes the possibility of taking initiatives, the right to suggest new matters for discussion, and the right to express an opinion of the discussed matter. The interaction structure, therefore, covers both the formal and the informal part of the organization.

The interaction structure suggested in conventional models projects IS professionals' interactions with users as a series of learning, analyzing, and suggesting protocols, followed by separate implementation and education phases. These protocols are started and controlled by the IS professional. The new administrative roles proposed have created a need for more flexible interaction structures. Boland [1978] discusses teaching, suggesting, and critiquing protocols, followed by trying and accepting phases. All these protocols mutually bind all IS developers. Mumford [1984] writes that participation implies mutual learning, which makes possible informed decision making, development of working relationships (mutual trust), and decrease of power domination [see also Høyer 1980]. She also points out that participation can differ with regard to the content of design, that is, which matters can become topics of interactions. In a similar vein, Keen and Scott Morton [1978] call for a broader scope of interaction, mutual commitment, and the building of realistic expectations.

#### 4.2.3 Interaction Strategies

An interaction strategy can be defined as those techniques that each participant or a

**Table 6.** IS Problems and the Impact of Development Organization Changes

(a) IS Development Process Problems

| Change | Goals | Technology | Economy | Process | View of organization | Self-image |
|---|---|---|---|---|---|---|
| Development organization | R | — | P | R | P | P |

(b) IS Use Process Problems

| Change | Operations | Data | Conceptual | People | Complexity |
|---|---|---|---|---|---|
| Development organization | — | P | R | P | — |

participant group can use to fulfill his or her role effectively. Traditionally IS professionals' interaction strategies have been characterized by neutrality, value freedom, and a commitment to apply technical skills correctly. Users, on the other hand, are expected to express their needs openly when asked.

The changes in administrative models and interaction structures have lead to proposals for new interaction strategies. Keen and Scott Morton [1978] stress that the designer must apply several tactics toward developing a contract for change. These include diagnosing possible resistance and selecting "countercounterimplementation" strategies [Keen 1981b]. Ciborra and Bracchi [1983] disclose the inevitability of opportunism in systems design. Each attempts to use strategies that exhibit information favorable to them and disregards other strategies. Boland [1979, 1981] points out that the IS development process implies an interpenetration with the users through language that is alien to the users. This creates the risk of intrusion. The IS professional must be aware of the boundary at which interpenetration turns into intrusion, for the latter lowers one's capability of being a good designer.

### 4.3 Summary

Development organization improvements focus on organizational features of the IS development environment: roles, communication patterns, and strategies. The strength of this body of IS research is its sensitivity to the social texture of the IS development process. Thus the improvements in the IS development organization can decrease IS problems that have a social origin and background (Table 6).

Unfortunately, there are few empirical studies on the positive impact of the changing development organization. Studies by Ives and Olson [1981] and Hirschheim [1983, 1985] show that the changing roles of IS developers produce some positive outcomes, but, in general, the results are inconclusive. Furthermore, it is difficult to establish and maintain new organizational arrangements. An empirical study by DeBrabander and Thiers [1984] shows detailed evidence that more relaxed interaction patterns strengthen the possibility for IS success.

## 5. MODELING IMPROVEMENTS

Changes in the development methods suggest that IS processes can be improved by creating models of the IS and its environments. These models can help to analyze, predict, and communicate the structure and content of the information system and its interactions with IS environments more accurately and completely than previously. In general, we classify IS development methods into three groups: IS models, IS environment models, and IS context models.

*IS models* represent information systems as a deterministic, technical system. *IS environment models* describe ambiguous, more malleable user and organizational environments and their interactions with the information system. Some IS development methods thus mix representations of the information system and some of its key environments. *IS context models* focus on several components of the IS context, including external, organizational, user, and development environments.

### 5.1 Information System Models

This method group focuses on the structure and behavior of the information system. The methods aim at an abstract, device-independent specification of the information system that can be used later on to implement the system in a specific operations environment. In this group we distinguish between the following development approaches:

- formalistic information modeling and
- functional.

#### *5.1.1 Formalistic Information Models*

Formalistic information models aim at a formal and nonalgorithmic specification of the IS [ANSI/SPARC 1975; Jardine and Reuber 1984; van Griethuysen 1982]. The static structure of the information system is captured by expressing data meaning in terms of entities, properties, and relationships [Flavin 1981; Rochfeld and Tardieu 1983; Tsichritzis and Lochovsky 1982; Wiederhold 1977]. The IS dynamics are expressed as triggers and events that change the data [Gustafsson et al. 1982; Jackson 1983]. Abstractions, assumptions, and constraints are also included in the formal IS model [Bubenko 1980; Flavin 1981; van Griethuysen 1982]. In the ideal case the IS model is isomorphic to some aspect of reality [Borgida 1985].

The formalistic information models are founded on axiomatic set theory [Stamper 1984], first-order theory [van Griethuysen 1982], or other logic systems [Furtado 1985; Stachowitz 1985]. They also provide user-friendly graphical languages and can be often augmented with computer-aided tools [Berild et al. 1984; Kung and Sølvberg 1984; Olle et al. 1982]. Thus, in contrast to functional approaches, they are geared toward more abstract specifications [Bubenko 1983]. In particular, they are valuable in an early formalization of the data [Furtado 1983]. They also support a more disciplined data analysis [Kahn 1982; Yao et al. 1982] than functional models and help to foresee the impact of data formalization on the organizational environment [Berild et al. 1984]. The completeness and consistency of the system specifications are easier to validate; specifications become more modularized, and they are not overloaded with technical detail. In this way formal information models promote systems designs that are more stable and can accommodate changes in the operations environment.

Formalistic information models have limitations, however. They disregard the characteristics of the users and the uses of data and their impact [Stamper 1985]. Although information models attempt to capture the meaning of data, they are inadequate in this task because they are based on a too naive theory of meaning [Winograd and Flores 1986]. In effect, information models have a bias toward treating "meaning" as a stable and immutable entity originating from outside human judgment [Kent 1978]. This bias can increase bureaucratic side effects and introduce dysfunctions in the information systems' processes [Lyytinen 1987a; Stamper 1986].

#### *5.1.2 Functional Information System Models*

The functional IS models are process oriented. They originate from the design of systems for processing documents represented in files of records. Functional models describe IS in terms of connected activities that process, transform, store, access, and modify data [DeMarco 1978; Ross and Schoman 1977; Ross 1977, 1985; Lundeberg et al. 1981]. Thus a modeling outcome is a representation of systems of activities with input–output relations to other systems. The mathematical basis of functional models is graph theory, especially Petri nets [Peterson 1977]. Functional models are also supported by appropriate computer-aided tools [Teichroew and Hershey 1977].

Functional models improve the quality of information system specifications [Chapin 1979; DeMarco 1978]. They specify more consistently and completely

**Table 7.** IS Problems and the Impact of IS Model Changes

(a) IS Development Process Problems

| Change | Goals | Technology | Economy | Process | View of organization | Self-image |
|---|---|---|---|---|---|---|
| Information models | — | P | P | R | P | — |
| Functional models | — | P | P | R | P | — |

(b) IS Use Process Problems

| Change | Operations | Data | Conceptual | People | Complexity |
|---|---|---|---|---|---|
| Information models | P | R | P | — | P |
| Functional models | R | P | P | — | P |

functions of the system regardless of a specific implementation strategy. In this way developers can grasp the dynamic behavior of the system and its functional relationships with the rest of the organization. This can also make it easy to predict its performance.

In spite of these advantages, functional models have limitations. They entertain a mechanistic view of the information system use process [Kensing 1984] and provide only little support for problem solving and organizational change [Kensing 1984; Vitalari 1984]. Furthermore, their view of the information system is too limited. They are abstracted from the physical implementation of the operations process, but neglect a deeper analysis of the social meaning of these processes.

#### 5.1.3 Summary

A key assumption in the IS model improvements is that the accuracy and correctness of IS representations have a positive impact on the IS development and use processes. The more consistent and complete the IS model is, the more efficient and effective the IS processes will be. Unfortunately, there is little empirical support to this claim. Very few, if any, systematic studies have been carried out in an attempt to understand the impact of IS development methods on IS processes. Floyd [1986] reports a series of cases in which several methods were used to solve the same application problem. The study concentrated only on IS development process problems. Her conclusions were that all studied methods had several limitations: They provided little help for problem solving, communication, and design tasks, and they made restricting (and unrealistic) assumptions about the development process, organization, and target systems. The same conclusions were reached in Vitalari's [1984] study, which demonstrates that system development methods support only slightly early development tasks that concentrate on goals and on the conceptual understanding of the problem. Thus development method improvements, as shown in Table 7, do not relate to all IS problems. In contrast, IS models seem to be applicable mostly to solving operations, data, and complexity problems. They may also have a positive impact on IS development process features and technological problems.

### 5.2 Information System Environment Models

IS environment models describe the interactions of the information system with one or several IS environments. Their primary aim is to understand how the IS environments interact and are affected by the information system and the qualities that they acquire through the interactions. Usually the focus is on understanding the IS use process, although some models shed light on the IS development process.

In IS environment models we distinguish among several method classes:

- information system architecture,
- information need,
- success factor,
- sociotechnical, and
- evaluation.

#### 5.2.1 Information System Architecture Models

Information system architecture models represent the overall IS architecture for an organization. The IS architecture identifies major applications and their principal linkages, and helps to manage, develop, represent, and assess an organization's application portfolio [Davis and Olson 1985; Kay et al. 1981; Martin 1983]. Thus the focus is on interactions between the information system and the organizational environment on an organizational level. The IS architecture is a normative concept: The current IS portfolio is compared with a future vision of what the portfolio should look like. The basic premise is that an organization needs the orderly growth of the application portfolio to operate effectively, and that growth need can be derived from features of the organizational, user, operations, and external environments.

Several methods have been proposed for developing an IS architecture. These include business systems planning [IBM, 1981; Martin 1983; Sullivan 1985], critical success factors [Rockart 1979; Sullivan 1985], King's strategy set transformation [King 1978], Davis and Wetherbe's long-range information systems planning [Davis and Olson 1985], McFarlan and McKenney's strategic grid approach [Davis and Olson 1985; Ives and Learmonth 1984], and Blumenthal's [1969] taxonomic framework for developing an information systems architecture. Some of these are similar to those used in identifying the information needs for a singular application. The contents of methods vary greatly in terms of focused IS environments. Some concentrate mainly on features of the organizational environment (goals, tasks, activities, and processes) [IBM 1981]; others place more emphasis on understanding the user environment and the user's perceptions [Rockart 1979].

The strength of IS architecture models is in observing the information system's role in improving the organization's total effectiveness, competitive situation, etc. These models also pinpoint a close connection between the corporate strategy and the information system's plan. A good IS architecture reduces the total expenditure of the IS function, improves the overall use of information systems, creates new services, and integrates the IS development process with organizational design.

Current IS architecture models have recognized deficiencies. Their view of organization is overly mechanistic. Their use can lead to increased bureaucracy and thereby neglect the potential of user-lead development, increase unnecessary systems development controls, and avoid risky, high-payoff development efforts. One crucial problem is that their validity cannot be easily detected. Furthermore, the models often tend to be either too complicated to be of any value in real life or too simple, making their credibility poor. Many have pointed out that IS architecture models are difficult to use and require considerable knowledge of the methods and business being analyzed.

The models seem to be relevant when dealing with goal problems, and partly relevant when dealing with economy, process features, and views of organizations. They are also instrumental in coping with data problems, and partly relevant with regard to conceptual, people, and complexity problems. There are few empirical assessments for using the IS architecture models. Some case studies have obtained positive results, however, when using critical success factor (CSF) approaches [Boynton and Zmud 1984; Rockart and Crescenzi 1984].

#### 5.2.2 Information-Need Approaches

Information-need approaches attempt to infer the need for an information system from the organizational, usually managerial, point of view. Emphasis is on understanding individual and organizational requirements for functions and information that the future information system ought to supply. Often, but not necessarily, information-need approaches are further refinements of the IS architecture model. Therefore, some development methods can be used in both areas.

A plethora of techniques, methods, and approaches have been suggested to accomplish the task of deriving information needs [Cooper and Swanson 1979; Davis 1982; Taggart and Tharp 1977; Yadav 1983]. These include normative analysis [Carlson 1979], decision analysis [Ackoff 1967], process analysis [IBM 1981], activity analysis [Lundeberg et al. 1981], critical success factors analysis [Rockart 1979], and strategy set transformation [King 1978].

The strength of information-need models is their emphasis on understanding the purpose of the IS use process in a larger context that embraces both the user and organizational environments. Thus they can be instrumental in solving relevant organizational or user's problems and in assessing alternative IS development options carefully.

Current need-oriented models have shortcomings that have received attention in recent literature. These relate mainly to assumptions that these methods make about the user and organizational environments. The need approaches have primarily an individualistic orientation and thus ignore salient interactions between the user and organizational environments [Ciborra 1984]. Furthermore, in understanding the user environment, the methods rely too much on the concepts of cognitive style and rational man in assessing the information need [Huber 1981, 1983]. In studies of the organizational environment overly cybernetic and mechanistic models are applied [Boland 1979; Ciborra 1984; Keen 1981b; Mitroff 1982], a nonconflicting concept of organization is presumed, and thus focus is only on some IS use processes [Goldkuhl and Lyytinen 1982; Hedberg and Jönssön 1978; Mason and Mitroff 1973].

Strikingly, although many methods for deriving information needs exist, there are few empirical assessments about their applicability and effectiveness. In addition, all such assessments consist of either short case studies or abstract feature analyses [cf. Cooper and Swanson 1979; Yadav 1983].

### 5.2.3 Success Factor Approaches

Success factor approaches study organizational and user environments, and their principal aim is to model factors like organizational characteristics, user-group features, individual differences, and prior computer experience to predict the IS implementation risks. In other words, the primary focus is on understanding how features in user and organizational environments determine the quality of the "fits" between the information system and its environments [Lucas 1981; Markus and Robey 1983].

Two lines of research have evolved to cope with IS implementation: theories regarding the organizational evolution of information systems and implementation risk factor models.

*5.2.3.1 Organizational Evolution Theories of Information Systems.* IS evolution theories disclose prevalent features in the information system and its environments that affect the quality of IS processes. We can discern two classes of theories: *growth models* and *frameworks of interrelated variables.*

*Growth models* [Nolan 1973, 1977, 1979] correlate IS problems with an S-shaped computing growth pattern predicted by an evolutionary teleological mechanism. The mechanism enables us to predict rises in computing costs by relating changes in the operations environment to an organizational learning curve describing the ability to control the technology diffusion. The learning curve is traced by changes in four growth processes: IS portfolio, the type of operations environment, the nature of the development environment (especially planning and control), and changes in the user environment (user awareness). Hence the model identifies complex interrelationships between organizational learning and technological development and their joint impact on IS problems.

Growth models offer a valuable heuristic device for studying and predicting the evolution of the IS function in all four areas. In fact, they have been one of the most widely used paradigms for managing organizational IS evolution. In particular, the growth models provide help in selecting appropriate strategies for advancing technology assimilation. These include finding technological discontinuities (rapid changes in the operation environments that may disrupt the whole IS context), identifying changes in user's awareness and skill, and helping IS applications and operations environment planning.

The growth models have weaknesses, however. One is that they tend to be too broad to be useful in dealing with specific IS problems. Their applicability in dealing with data problems, for example, is difficult to show. Another is that growth models include theoretical and empirical shortcomings: Their empirical base is questionable, they are unaware of their normative content, and finally they fail to account fully for the complex relationships among all components in the IS context [King and Kraemer 1984].

*Frameworks of interrelated variables* model the IS context by independent, partly independent, and controlled variables and attempt to predict the outcome of the dependent variables. Usually the selected variables cover all IS context environments and processes, but the primary focus is on the user and organizational environments. Various frameworks have been proposed by, among others, Mason and Mitroff [1973], Lucas [1975], Dickson et al. [1977], Bostrom [1978], Ein-Dor and Segev [1978], and Ives et al. [1980]. More specifically, Ein-Dor and Segev [1978] suggest 22 propositions as to how independent, partly independent, and controlled variables affect IS success. Another example is Ginzberg's [1980] study of interactions between system characteristics and organizational characteristics. These together determine an "organizational fit" affecting the IS implementation outcome.

The value of the research into variable frameworks is that it has stimulated investigations into the cause–effect patterns that underlie IS problems. These investigations have provided valuable insights into the nature of IS problems, suggested heuristics for solving them, and discussed aspects to consider in understanding the quality of IS processes. Unfortunately, the results obtained from these studies are inconclusive [Attewell and Rule 1984; Turner 1982]. Although new frameworks have been more general and detailed in their selection of variables [Ives et al. 1980], they have not been of much help in coping with IS problems. The reasons for their low practical utility are manifold: lack of an underlying theory, lack of general agreement on what variables to select, haphazard division of variables into independent and dependent ones, naive application of strict causal explanation of relationships among variables [cf. Checkland 1981], and so on.

In general, we lack empirical studies on how theories of the evolution of information systems influence IS problems. These theories tend to be general, and for this reason their applicability in dealing with particular IS problems is difficult to measure. The greatest gains, however, seem to be coming from a deeper understanding of process features, conceptual problems, and problems with people. Growth models may also help to reduce the complexity of the IS development process.

*5.2.3.2 Implementation Risk Factor Models.* Implementation risk factor models shed light on the dynamics of organizational and user adaptation to an information system. They can be grouped into three model classes. The oldest models reflected professionals' implementation experience. As Keen and Scott Morton [1978] show, the problem with these models is that they often suggest contradictory lines of action.

The second model type involves factor models [Alter 1980; Ginzberg 1981; Keen and Scott Morton 1978; Lucas 1981; Powers and Dickson 1973]. The factor paradigm attempts to answer the following question: What factors are associated with IS problems, especially with people's reactions to IS? Examples of would-be factors [Ginzberg 1975] are top-management support, felt need, and IS quality. Unfortunately, the factor models have been of little help in explaining the success in implementation. They lack an underlying theory; studies concentrate only on easily measurable variables and disregard factors that are difficult to measure [Keen 1974]; they show inconclusive patterns between selected factors and success in implementation [Turner 1982].

Some of the recent implementation studies have enlarged earlier factor models by incorporating a view of the implementation process into the implementation model [Ginzberg 1978]. The process views have been mainly derived from political and conflict theories of organizational change. A political model suggests [Kling and Iacono 1984; Markus 1983; Markus and Pfeffer 1983] that organizational power is an important factor to consider in IS development. Therefore, an investigation into the social and political climate affecting opportunities for successful IS intervention is needed. This investigation must disclose how organizational power balances are to be changed as a precondition for effective IS change. The conflict model [Franz and Robey 1984; Robey 1984] delineates the conflict resolution process involved in IS development. Two ways of resolving the conflict exist: constructive resolution and management by negotiated compromise. Robey [1984] noted that these strategies are largely culturally based and can be applied simultaneously to interpret the same sequence of development events.

The implementation risk factor models have been validated by extensive empirical research [Alter 1980; Keen and Scott Morton 1978]. They are relevant to an understanding of development problems in goals, economy, process features, view of organization, and self-image. They may also help in coping with problems with data, conceptual understanding, and people.

#### 5.2.4 Sociotechnical Approaches

Sociotechnical approaches adopt the perspective that IS development involves the design of a work organization. Thus they incorporate features of the information system, user, and organizational environments into a sociotechnical model of the IS context. Representatives of the sociotechnical approach are Mumford [1981, 1983], Bostrom and Heinen [1977a, 1977b], and DeMaio [1980].

In the sociotechnical approaches the information system has to be compatible with the surrounding social system, that is, the user and the organizational environments. This is achieved by developing social and technical alternatives simultaneously and comparing them with regard to their ability to meet both social and efficiency objectives [Land et al. 1980].

Sociotechnical approaches offer an effective strategy for dealing with many social implications of the IS change [Mumford 1983]. Many empirical applications of the method have shown that it allows a balanced adoption of the information system into user and organizational environments and takes into account prevailing social values and norms, elusive social needs, users' skills and competence, and so forth [Bostrom and Heinen 1977b; Hirschheim 1985; Land 1982].

However, sociotechnical approaches may sometimes suggest a too limited change strategy, as their basic image of an information system is limited to a technical system. Therefore sociotechnical approaches focus mainly on the IS operations process. Whenever guidelines are given on how to model the IS use process [cf. DeMaio 1980; Mumford 1983], they are based on a limited cybernetic outlook on the nature of knowledge. Moreover, sociotechnical approaches assume that consensus is a main feature of organizations as open systems. For this reason sociotechnical designs may fail to recognize the politics of systems design that relate to the shifts in the organizational power balance [Markus 1983]. Finally, sociotechnical designs assume a "two-track" development trajectory. The first, the technical track, can be carried out by the conventional IS development methods. The second, the social track, uses specific sociotechnical methods. However, how these two tracks, with different world views, can always be smoothly combined is not sufficiently addressed. For example, if technical goals and social needs cannot be balanced, there is no room to use the sociotechnical approach.

#### 5.2.5 Evaluation Approaches

The commonplace approach to IS evaluation has been to consider technical or cost-efficiency criteria, or to base it on intuition [Carlson 1974; Emery 1974]. This approach, in spite of its simplicity, has lead to the development of information systems that are technically sound, but nonetheless organizationally unacceptable or ineffective. Therefore, a search has been made for more elaborate evaluation methods.

Proposed evaluation approaches combine elements from the information system and its environments. These approaches suggest decision rules that may guide decision makers to a more informed and "rational" decision [Carlson 1974; Emery 1971, 1974; King and Schrems 1978; Kleijnen 1980].

An evaluation method distinguishes three elements for IS decision making [Klein and Hirschheim 1983; Welke 1978]:

(1) What is the impact of the information system on IS environments?
(2) How are these consequences measured?
(3) How can the measures be combined to yield a "good decision"?

The first question deals with the problem of improving representations of the IS context. Consequence measures then refine the IS context representations to determine the merits of proposed IS designs. On this level, Carlson [1974] distinguishes among six measurements: event logging, attitude survey, rating and weighting, systems measurement, systems analysis, and cost–benefit analysis. In the third step, the measures are combined by applying decision rules that state how the measures can be combined and compared.

The value of the evaluation frameworks is that they recognize a wider range of consequences and suggest more elaborate decision rules than intuitive or technical considerations. Their main thrust is on information system effectiveness [Emery 1974; Hamilton 1980; Welke 1978]. The measures refer to the accomplishment of an organization's objectives, reasons for their variance, and selection of alternatives that achieve them. This has resulted in large and complicated frameworks. For example, King and Schrems [1978] enumerate 26 possible IS benefits (performance measures) that can contribute to effectiveness. Hamilton [1980] suggests a framework that involves 12 objectives grouped into four levels, combining 35 performance measures. The measures range from improved time of information presentation to higher sales revenues.

The value of evaluation approaches is twofold. They identify the goal-oriented nature of IS design and the importance of "purposefulness" for the definition of rationality. Second, the evaluation approaches demonstrate the limitations of technological and cost criteria for effective IS design, and suggest the consideration of a wider range of possible consequences.

Evaluation approaches also have deficiencies, and often their practical use is limited. First, Welke [1978] criticized their way of combining consequence measures. He pinpoints the fact that most evaluation methods combine several measures to yield a single value on one scale. This approach is, however, too simple and leads to complicated measurement problems. Instead, Welke prefers the application of balances and heuristics. Second, in most frameworks, one calculates with "objective measures" to select the "optimal" alternative. In these frameworks one thus loses sight of the fact that goals, values, and measures are actually the results of a social negotiation [Bloor 1976]. Third, evaluation approaches assume that decision goals can be accepted by all the members of an organization. In an empirical study, however, Kumar and Welke [1984] showed that this is not the case, even within IS development groups. Fourth, the approaches ignore the fact that "rational" selection measures are often used to achieve some limited political goals [Franz and Robey 1984; Kling 1980; Mayntz 1984; Robey and Markus 1984].

#### 5.2.6 Summary

The essential point with regard to IS environment models is that an understanding of the effective IS use process requires a penetrating analysis of IS environments. Several development methods can help in this analysis (Table 8). IS environment models can help to overcome IS problems in data, conceptual understanding, and people's reactions. There is also some empirical evidence to support these claims, although few studies have been carried out to test them systematically.

**Table 8.** The Impact of IS Environment Models on IS Problems

(a) IS Development Process Problems

| Change | Goals | Technology | Economy | Process | View of organization | Self-image |
|---|---|---|---|---|---|---|
| Information system architecture | P | P | P | — | P | — |
| Information-need approaches | P | — | P | P | R | — |
| Success factor approaches | R | — | P | R | P | P |
| Sociotechnical approaches | R | — | P | R | R | — |
| Evaluation approaches | R | P | R | P | P | — |

(b) IS Use Process Problems

| Change | Operations | Data | Conceptual | People | Complexity |
|---|---|---|---|---|---|
| Information system architecture | P | R | P | — | P |
| Information-need approaches | — | P | R | — | — |
| Success factor approaches | — | P | P | R | — |
| Sociotechnical approaches | — | — | P | R | — |
| Evaluation approaches | P | P | R | P | P |

However, the impact of IS environment models has not been as pervasive as expected, and many problems are associated with their use. First, their adoption into practice is not easy and involves large investments in education [Hirschheim 1985]. Second, their use may necessitate considerable overhead in systems development, the outcomes are not always clear, and often they cannot be related easily to other methods. Furthermore, IS environment models may require drastic changes in the organization and process structure aspects of the IS development environment that may be difficult to achieve because of prejudices, vested interests, or resistance to change. Third, IS environment models tend to emphasize formality and causal explanation [Boland 1979; Weick 1984] in understanding user and organizational behaviors. There is a considerable dearth of theories that do not adhere to such a mechanistic view.

### 5.3 Information System Context Models

IS context models focus simultaneously on several IS context components. These may include features of the operations, user, organizational, and development environments, as well as the type of information system being developed. Their principal goal is to decide on changes in one of the IS processes by determining an appropriate mix of changes in other IS context components. Usually the focal point is to determine an appropriate context for a specific type of IS use process (organizational IS contingency theory; see Olerup [1980]) or IS development process (IS development contingency theory). The former models are rare in the IS literature. Therefore we shall discuss only the latter class of IS context models.

#### 5.3.1 Information System Development Contingency Models

The IS development contingency models are based on the conviction that every IS development process is different. Therefore the development process should not be standardized. Instead, a specific systems development strategy should be chosen on the basis of prevailing "contingencies." The identification of these contingencies can help in selecting an effective mix of tools, modeling techniques, process strategies, and organizing principles. The general form of contingency models is as follows: If contingencies C(1), C(2), . . ., C($n$) prevail, then the IS development strategy should have features F(1), F(2), . . ., F($m$).

Examples of contingency models are Davis's [1982] framework for choosing among information requirements determination strategies, Ahituv and Neumann's [1984] framework for planning the systems

**Table 9.** The Impact of IS Development Contingency Models on IS Problems

(a) IS Development Process Problems

| Change | Goals | Technology | Economy | Process | View of organization | Self-image |
|---|---|---|---|---|---|---|
| Contingency models | P | — | P | P | P | P |

(b) IS Use Process Problems

| Change | Operations | Data | Conceptual | People | Complexity |
|---|---|---|---|---|---|
| Contingency models | P | P | P | P | P |

life-cycle activities, Iivari's [1983] sociocybernetic systems development model, and Swanson's [1984] framework for organizing programming maintenance process.

The antiuniversalist idea of contingency approaches agrees with the common wisdom of systems development, and contingency frameworks are taking steps toward more adaptable development strategies and reminding developers to consider the interdependencies among several areas of systems development: development methods, process structures, organization, etc. In this sense, contingency models have some relevance in solving all sorts of IS problems (Table 9).

There are several limitations with regard to how contingency frameworks are being developed and applied. First, contingency studies in other areas, such as organization design [Mayntz 1984] and decision theory [Huber 1983], have been inconclusive. There is no evidence that they would be more satisfactory in the IS area. Second, current contingency frameworks are fragmentary, and the choice of contingency factors suffers from the same problems found in success factor models: an emphasis on measurable aspects, a strictly causal explanation scheme, neglect of cultural environment, the ad hoc nature of frameworks, and much ambiguity when applying the frameworks to specific development processes. There is also a dearth of empirical studies that would validate the applicability of contingency models.

## 6. THEORY IMPROVEMENTS

Many suggested changes in development methods, organization, and process structures have been based on a fresh theoretical look at the IS context. Therefore, developments in more abstract theories and models about the IS context are important for the IS community's ability to recognize, analyze, and solve IS problems. In this section we discuss a number of recent developments in IS theories. These theories provide a foundation for several of the IS development methods, techniques, and approaches discussed in Sections 2–5.

The traditional theories about the IS context are founded on two often overlapping sets of notions: those that relate information systems to technical artifacts and those that relate the IS use process to managerial decision making. These theories have been used mainly to describe and explain the phenomena in the IS context and particularly the qualities of the IS processes.

According to the technical theory, an information system is a complex technical artifact that can be improved by better engineering [Davis and Olson 1985; Senko 1975, 1977]. The focus here is on the relationships between the information system and the operations environment. An information system consists of processors, data stores, data communication devices, software, procedures, and people. According to the decision theory an information system supports rational managerial decision making, operations, and planning processes [Ackoff 1967; Ein-Dor and Segev 1978]. The focus here is on interactions among the information system, the user, and the organizational environments.

During recent years the technical approach has been confronted with the sociotechnical and class-conflict views, in which the classical approach is extended by a more encompassing view of the IS operations process as a work process. The decision-oriented view has been augmented or challenged by the inquiry, sense-making, soft-methodology, contractual, and language-action views. These suggest alternative theoretical approaches to delineate and understand the interactions among the information system, the organizational environment, and the user environment.

### 6.1 Alternatives to the Technical View

#### 6.1.1 Sociotechnical Theory

Sociotechnical theory serves as a theoretical foundation for the sociotechnical method explored in Section 5.2.4, and for many recent advances toward understanding the participatory development organization (Section 4.2.1) and the pluralistic development structure (Section 3.3.1). It was originally developed in the Tavistock Institute by such scholars as Emery [1972] and Trist [1981].

The sociotechnical theory extends the closed (technical) systems view of the information system with a theory of organizations as open, adaptable systems. In other words, it addresses the indispensable interactions of the technological core of the information system with the user and organizational environments. Information systems are open sociotechnical systems, and IS processes (especially the operations process) involve technology that transforms raw materials into output and a social structure that links the human operators with both the technology and each other. The purpose of sociotechnical theory is to combine these subsystems with the IS development and use processes so that technical and social goals are achieved and reinforce the achievement of the overall goals of the whole system (organizational adaptability, survival).

#### 6.1.2 Class-Conflict Theory

The class-conflict theory recasts the idea of technology as an end in itself. On the basis of the Marxist analysis of society and organizations [Burrell and Morgan 1979; Marx 1976], the IS process is viewed as a work process. However, it disagrees with the sociotechnical theory as to what constitutes the key characteristics of this process. Class-conflict theory argues that all work processes are characterized by a fundamental contradiction between capital and labor and by the fact that this contradiction affects the ways in which they are organized. Therefore, the impact of IS use and development processes is not neutral [Ehn and Sandberg 1979], and IS developers and researchers must become aware of this impact and consciously select a group (class) that is affected by these processes [Ehn and Sandberg 1979].

The class-conflict theory has motivated advances in some new development methods, process structures, and development organizations, especially in Scandinavian countries [Sandberg 1979, 1985], Great Britain [Rosenhead and Thunhurst 1982], and West Germany [Briefs 1980; Kubicek 1983]. It replaces the traditional IS development strategy with a trade union alternative that aims at the democratization of working life. This implies both a change in the conditions for introducing and applying technology to the work process and changes in organizations' planning and control mechanisms.

### 6.2 Alternatives to the Decision View

#### 6.2.1 Inquiry Theory

The inquiry theory focuses on the IS use process, which is seen from the viewpoint of individual decision making. This is done in an insightful way, which considerably extends traditional decision theory and sheds light on varying types of inquiries available to an individual when confronted with a decision problem. The inquiry theory originates from Churchman's [1971] work on inquiring systems. The inquiry theory has also been influenced by studies on cognitive psychology [Jung 1931] and by Anthony's [1965] work on management control levels. The most prominent researchers into the inquiry theory have been Mason and Mitroff [Mason and Mitroff 1973; Mitroff and Mason 1983].

Investigations into cognitive style indicate that IS users reason differently according to their cognitive type. In these studies, the IS user's cognitive habits and strategies are categorized on a fairly broad level, and such behavior is essentially viewed as a personality variable [Keen and Scott Morton 1978]. The conventional decision view emphasizes mainly one cognitive type (thinking–sensation) in the Jungian typology. In the inquiry theory it is hypothetized that this leads to the ineffective use of information systems.

The IS use process is also affected by the epistemological strategies preferred by the individual. Epistemological strategies are classes of evidence generators or guarantors. Each evidence generator exhibits an archetypal way of generating evidence for decision making. Following Churchman's classification, Mason and Mitroff [1973] suggest five evidence-generating strategies: Lockean, Leibnitzian, Kantian, Hegelian, and Singer/Churchmanian. They note that the decision-oriented view recognizes only two of them: Lockean inquiry (by its emphasis on the databases and "enterprise descriptions") and Leibnitzian inquiry (by its emphasis on the management science/OR models). From this viewpoint, the decision strategy attempts to force all IS use processes to conform to just two evidence guarantors.

The importance of Mason and Mitroff's work lies in their cogent argument that the received decision theory is too limited. This has initiated insightful research into the impact of user environments and the inquiring strategies that IS users employ. Therefore, many improvements in the development methods concerned with IS environments, changes in the development organization (interaction strategies), and deeper understanding gained from the IS development process as an inquiry and bargaining process are founded on the inquiry theory. However, many original insights offered by the Mason and Mitroff research program have not been fully realized. For example, few empirical studies have explored the role of inquiring strategies in the IS use process [cf. Chua et al. 1981; Ulrich 1983].

#### 6.2.2 Sense-Making Theory

The sense-making view explores the IS use process as a social process by which individuals interpret situations and construct their actions. Therefore, the focus here is on individuals' interactions that are affected by the IS use process, that is, the interactions between the information system and the user and organizational environments. The theory is rooted in symbolic interactionism [Blumer 1969; Burrell and Morgan 1979], ethnomethodology [Burrell and Morgan 1979; Garfinkel 1967], and sociology of knowledge [Berger and Luckmann 1967; Wuthnow et al. 1984].

The basic premise of the sense-making theory is that people act on the basis of the meaning that they attribute to situations. Action is not a mere release in response to some predefined presented stimuli, but emerges from social interaction and is developed and modified through an interpretive process.

The IS use process is a part of the social environment within which users interact to develop meanings and interpretations of an inherently ambiguous reality. An information system stands as one of the major available means of ordering and interpreting users' experience. Symbols in an information system can serve both means–ends rational and interpretive tasks [Boland and Pondy 1981]. Information systems are means–ends rational to the extent that they convey measurable characteristics of the organization; they are interpretive tools to the extent that their categories impose coherence on a chaotic organizational process by "defining" what is real and bringing it to the attention of the management process. Only the former process is recognized in the decision view.

In the IS area, sense-making theory has been advocated by such scholars as Boland [1979, 1984, 1985] and Goldkuhl [1980]. New ways have been suggested for structuring the IS development process [Boland and Day 1982] and developing IS environment models [Boland 1979; Goldkuhl 1980], and this has motivated some studies into the nature and content of interactions during the IS development [Boland 1978]. In spite of its originality, however, the impact of sense-making theory on the IS community has been modest. One reason for this is that there are few practical proposals on how to implement it. Also, with the exception of Boland's work [1978, 1984], empirical studies on the impact of using a sense-making theory-based approach are lacking.

*6.2.3 Soft-Systems Methodology*

Soft-systems methodology was developed by Peter Checkland [1981, 1984] and his colleagues at the University of Lancaster. Its impact is visible in many European methodologies (see Episkopou and Wood-Harper [1984]). This methodology results from applying a systems engineering approach to the solution of real-world problems in action research settings [Checkland 1981]. By a real-world problem, Checkland means deficiencies in the operation of human activity systems, that is, in systems of which a human being is a part and in which human intentions play a decisive role. The methodology focuses mainly on problem conceptualization and modeling tasks that deal with user and organizational environments.

In contrast to the modeling principles applied in operations research or systems engineering, several valid representations (models) of the organization are available in soft-systems methodology. Each one flows from a different world view, or weltanschauung. The role of a weltanschauung is to provide a horizon to account for and appreciate the problem. Checkland claims that the adoption of a horizon is typical of all problem solving in human activity systems. Problems can (and must) be grasped from several (conflicting) weltanschauungs, where each one interprets the situation differently. Therefore we cannot speak of models as being *correct*, but as being *meaningful.*

The soft-systems methodology helps us to appreciate the relevance of the IS use process to the organizational and user environments. The methodology has motivated advances in some development methods [Checkland 1981], focusing on user and organizational environments. It has also had a considerable impact on ways of thinking about the development process as a bargaining and discourse process. It puts emphasis on open development organizations that strengthen participation and learning. There are many reports of case studies in which the effective use of the soft-systems methodology in real-world settings is described [Checkland 1981].

*6.2.4 Contractual Theory*

The contractual theory is based on the transaction cost theory of organizational behavior [Ouchi 1979, 1980; Williamson 1975, 1979; Williamson and Ouchi 1981]. The focus here is on the interactions between the organizational environment and the IS use process. In the IS field the major contributor to contractual theory has been Ciborra [1981, 1984].

The contractual theory centers around the notions of organizational *exchange* and *contract.* An organizational exchange is a process in which the agents involved anticipate prospective gains. Contracts are a means of defining and controlling the exchanges. Finally, agents need an information system, that is, network of information flow and stock, to create, set up, control, and maintain the contracts.

The theory suggests that organizations differ owing to the complexity and ambiguity of the exchanges [Williamson 1975]. Thus the organizational information systems needed to create and monitor contracts vary. Extremes in contracting complexity are "spot contracts" taking place in an open market and "authority relations" created, for example, by hiring an employee for an unlimited period of time. In the former situation the exchange ambiguity is minimal, whereas in the latter it is extremely great. Thus, in hiring an employee, the involved participants cannot predict all the contingencies of the contract. The need to cope with the contract contingencies explains the development of two organizational forms, market and hierarchies, and the evolution of different types of information systems. This can be exemplified by the development of information systems supporting stock exchanges or payroll systems in markets and hierarchies, respectively.

Another new aspect in the contractual theory is the level of potential exchange opportunism. Contracting always involves two rational participants whose goals may conflict. Opportunism arises from the benefits of safeguarding the contractor's interests with guile, that is, making false or empty threats and promises. One reason for developing information systems is to use them for opportunistic purposes or for increasing the participant's knowledge of the contracting process so that he or she can better defend his or her self-interest. Examples of information systems that reduce chances for opportunism are credit-control systems or client-tracking systems.

The impact of the contractual theory on the IS context has been modest thus far. One example of the growing interest in this theory is Beath's [1983] work, in which she applies contractual theory to analyzing systems development organizations. Also some advances in IS environment models, in which the role of information systems in reframing the organization's competitive situation is discussed [Ives and Learmonth 1984] or information needs from the study of "orders" are derived [Carlson 1979], can be theoretically explained by the contractual theory.

### 6.2.5 *Language Action Theory*

The language action theory has its origins in linguistics [Wünderlich 1979], philosophy of language [Searle 1969, 1979; Wittgenstein 1958], and social theory [Habermas 1979]. It focuses on the relationships between the user and organizational environments when the interactions are observed as forming a linguistic process: an exchange of linguistic utterances between organizational actors.

The language action theory combines ideas from the contractual and the sense-making views. It is based on the contractual view to the extent that it emphasizes the contractual base of the IS use process; it is based on the sense-making view to the extent that it emphasizes the role of the interpretive process in understanding the IS use process.

A distinguishing feature of language action theory is its focus on the IS use process as a process of communicative action. Communicative action is carried out through communicative acts, which are minimal units of human communication. A communicative act is performed when producing (by writing, speaking, or other signals) a comprehensible message with a genuine intention in a context. Information systems, in the same vein, are intentionally arranged formal linguistic systems that support, enable, control, or coordinate people in their action. They involve well-defined communicative acts, where each communicative act has a specific intention related to the accomplishment of organizational tasks (contracting and exchange).

There is a growing interest in studying the IS use process as a linguistic process [De Cindio et al. 1986; Flores and Ludlow 1981; Goldkuhl and Lyytinen 1982, 1984; Kimbrough et al. 1984; Lehtinen and Lyytinen 1986; Lyytinen 1985, 1987b; Mathiassen and Andersen 1983; Winograd and Flores 1986]. Language action theory has been mainly applied to the derivation of new IS environment models [De Cindio et al. 1986; Lyytinen and Lehtinen 1984a, 1984b]. It has had much less impact on other IS context components. Goldkuhl and Lyytinen [1984], however, discuss some implications of the language action theory for the development organization and interaction strategies. Some developments in so-called collective contracting tools in the operations environment [Sluizer and Cashman 1984] are also informed by the language action view.

## 6.3 Summary

We have discussed seven IS theory improvements; it is assumed that IS problems are partly a result of the current inadequacy of the IS theory. Therefore prevailing theories should be expanded and/or replaced by better ones.

Emerging theories offer original insights, and their heuristic value is considerable. As Table 10 shows, they can help in dealing with IS development problems associated with goals, process, view of organization, and self-image. New theoretical insights on the IS use process will mainly affect problems with data, concepts, and people.

However, there are great difficulties in applying new IS theories, and their rate of

**Table 10.** IS Problems and the Impact of New IS Theories

(a) IS Development Process Problems

| Change | Goals | Technology | Economy | Process | View of organization | Self-image |
|---|---|---|---|---|---|---|
| Sociotechnical theory | P | — | P | R | R | R |
| Class conflict theory | R | — | — | R | R | R |
| Inquiry theory | P | — | — | R | — | R |
| Sense-making theory | R | — | — | R | P | R |
| Soft-systems methodology | R | — | — | R | P | R |
| Contractual theory | R | — | — | R | P | R |
| Language action theory | R | — | — | R | P | R |

(b) IS Use Process Problems

| Change | Operations | Data | Conceptual | People | Complexity |
|---|---|---|---|---|---|
| Sociotechnical theory | — | — | R | R | |
| Class conflict theory | — | — | R | R | — |
| Inquiry theory | — | P | R | R | — |
| Sense-making theory | — | P | R | R | — |
| Soft-systems methodology | — | P | R | P | — |
| Contractual theory | — | P | R | P | P |
| Language action theory | — | R | R | P | — |

adoption into practice has been low. There are several reasons for this. First, new theories are usually offered as a by-product when accepted tradition is criticized. There is less research that seriously develops frameworks, methods, and tools. Hence an evaluation of the usefulness of the new theories is difficult, and there are few attempts in this direction. Thus we cannot make any detailed theory comparisons. Another problem is that currently we do not have any framework for relating IS theories to each other. This makes their selection and combined use difficult. Third, some of the emerging theories suggest development approaches that are too radical or different from current methods of developing systems.

## 7. CONCLUSIONS

### 7.1 Summary

We have made a rapid tour around the IS research literature dealing with IS shortcomings. We have also surveyed what research results have been suggested to remove them. Our study shows that IS researchers and practitioners are becoming increasingly aware of the multitude of IS problems. Many IS researchers are concerned with understanding connections between IS problems and features of the IS context, with probing effective cause and effect patterns, and with applying new perspectives to the study and solutions of IS problems.

The main results of our survey are summarized in Tables 11 and 12, which indicate how changes in the IS context primarily affect IS problems.

First, it is suggested in the tables that IS context changes can contribute to dealing effectively with all IS problems. All IS shortcomings could be decreased by appropriate education, new development methods, technology investments, and organizational rearrangements. In practice, however, the rate and severity of IS problems show no substantial decrease. How can this be so, if methods for rectification are known and available?

In our opinion, the most important reason is that the improvement of the IS processes involves a complex undertaking, and as a result IS problems cannot be reduced within a short period of time. Most causes of problems, as our study shows, are social in nature, and their removal may require intricate learning and organizational changes, which are slow in coming. In addition, the nature and roots of problems are unclear, and their relationships are complex. This makes it difficult to understand the problems and their connections properly and complicates the choice of appropriate measures. Furthermore, even if the

**Table 11.** A Summary of IS Development Problems and the Impact of IS Context Changes

| Changes | Section | Goals | Technology | Economy | Process | View of organization | Self-image |
|---|---|---|---|---|---|---|---|
| Computing environment | 2.1 | — | R | P | P | P | — |
| Support environment | 2.2 | — | — | R | P | — | — |
| Application generator | 2.2 | — | — | R | P | P | — |
| Life cycle | 3.1 | — | R | P | P | — | — |
| Prototyping | 3.1 | P | P | P | R | P | — |
| PSC | 3.1 | P | P | R | R | P | — |
| Evolutionary approach | 3.2 | P | — | P | R | P | — |
| Organizational change | 3.2 | P | — | P | R | P | — |
| Bargaining | 3.3 | R | — | P | R | P | R |
| Discourse | 3.3 | R | — | — | R | P | R |
| Development organization | 4.1 | R | — | P | R | P | P |
| Information models | 5.1 | — | P | P | R | P | — |
| Functional models | 5.1 | — | P | P | R | P | — |
| Information system architecture | 5.2 | P | P | P | — | P | — |
| Information-need approaches | 5.2 | P | — | P | P | R | — |
| Success factor approaches | 5.2 | R | — | P | R | P | P |
| Sociotechnical approaches | 5.2 | R | — | P | R | R | — |
| Evaluation approaches | 5.2 | R | P | R | P | P | — |
| Contingency models | 5.2 | P | — | P | R | P | P |
| Sociotechnical theory | 6.1 | P | — | P | R | R | R |
| Class conflict theory | 6.1 | R | — | — | R | R | R |
| Inquiry theory | 6.2 | P | — | — | R | — | R |
| Sense-making theory | 6.2 | R | — | — | R | P | R |
| Soft-systems methodology | 6.2 | R | — | — | R | P | R |
| Contractual theory | 6.2 | R | — | — | R | P | R |
| Language action theory | 6.2 | R | — | — | R | P | R |

**Table 12.** IS Use Process Problems and the Impact of IS Context Changes

| Change | Section | Operations | Data | Conceptual | People | Complexity |
|---|---|---|---|---|---|---|
| Computing environment | 2.1 | R | P | — | P | P |
| Support environment | 2.2 | R | P | — | — | R |
| Application generator | 2.2 | R | P | P | P | P |
| Life cycle | 3.1 | R | P | — | — | R |
| Prototyping | 3.1 | R | P | P | P | P |
| PSC | 3.1 | R | P | R | P | R |
| Evolutionary application | 3.2 | — | P | R | P | — |
| Organizational change | 3.2 | — | — | P | R | — |
| Bargaining | 3.3 | — | P | R | R | — |
| Discourse | 3.3 | — | R | R | R | — |
| Development organization | 4.2 | — | P | R | P | — |
| Information models | 5.1 | P | R | P | — | P |
| Functional models | 5.1 | R | P | P | — | P |
| Information system architecture | 5.2 | P | R | P | — | P |
| Information-need approaches | 5.2 | — | P | R | — | — |
| Success factor approaches | 5.2 | — | P | P | R | — |
| Sociotechnical approaches | 5.2 | — | — | P | R | — |
| Evaluation approaches | 5.2 | P | P | R | P | P |
| Contingency models | 5.3 | P | P | P | P | P |
| Sociotechnical theory | 6.1 | — | — | R | R | — |
| Class conflict theory | 6.1 | — | — | R | R | — |
| Inquiry theory | 6.2 | — | P | R | R | — |
| Sense-making theory | 6.2 | — | P | R | R | — |
| Soft-systems methodology | 6.2 | — | P | R | P | — |
| Contractual theory | 6.2 | — | P | R | P | P |
| Language action theory | 6.2 | — | R | R | P | — |

problems and appropriate measures are known, the outcome may still be uncertain as we do not know enough about the mechanisms that produce IS problems and how the measures will affect them. As observed in our survey, there is a critical shortage of empirical studies into how different changes contribute to solving different problems. Finally, the IS research always lags behind what is happening in practice, and we have the problem of a "moving target." Often appropriate research results are not available when a particular type of IS problem occurs.

Second, there is the "joint impact" problem: Measures are rarely adopted separately; instead, changes in the IS context come in packages. However, there are few systematic studies on the connections between suggested measures. Hence it is difficult to show which individual changes are instrumental in solving a particular problem and how different measures can be combined.

Third, there are several measures for tackling each problem class. The classification scheme employed here does not, however, show to what extent different measures can, indeed, solve a problem and for which types of problems a particular measure is most appropriate. For example, are information modeling approaches better for solving complexity problems than functional approaches, or is the impact of supporting environments greater when compared with improvements in development methods? In this paper we have not made such comparisons because there simply are not enough empirical data available. It seems likely, however, that each solution requires a flexible combination of several IS context components [see Ahituv and Neumann 1984; Berild et al. 1984; Davis 1982]. The mixture varies from one situation to another, but at this stage there is paucity of knowledge as to how to select an effective mixture of measures for each type of problem. To exemplify, complexity problems may be tackled by both functional and information models, and by developing appropriate development support environments around the use of these methods. In another situation, however, a different mix may be more appropriate, for example, simplifying the IS application.

Fourth, proposed IS context changes can be divided into two larger groups. Changes in the operations environment contribute to improvements in economy and efficiency. They also introduce new types of information systems—a factor not considered in depth in this paper. These changes can be achieved by sufficient reinvestments in technology and education, although the shortage of educated people may slow down the introduction of new technology. Changes in the IS development environment have a wide impact on the whole spectrum of IS problems. These changes are, however, difficult to implement, because they involve learning and social change. Therefore, improving the productivity and effectiveness of the IS context is a question of appropriate managerial and organizational IS policy.

### 7.2 Discussion

What are the implications of our journey? Two sets of implications are briefly reviewed here: research implications and implications for practice.

Our survey shows several weaknesses in IS research dealing with IS problems. Four shortcomings are discussed below: weaknesses due to theoretical diversification of the field, research core assumptions, conceptual limitations of the research, and lack of empirical research.

Our survey shows that, in order to attack IS problems effectively, researchers must grasp many theoretical approaches and apply several research strategies. Unfortunately, this sort of pluralism is not in fashion because academic research rarely gives proper credit to pluralism. Therefore IS research is fragmented and mostly uncoordinated, and cumulative research traditions are rare.

Many IS researchers rarely ponder core assumptions that define the validity and relevance of conducted research and the nature of the problems that it attempts to solve. Instead, IS researchers often view IS problems from the viewpoint of their own idealistic research pattern. This results in a problem–method gap: the research methods are not related to the types of problems, and they tend to appeal to the rational ideal of research [Mumford et al. 1985]. Each IS problem, however, may require its own type of theory building and research.

IS researchers often are not conscious of the conceptual limitations of their research approach. As a result, the limitations of the proposed measures are too rarely evaluated critically. This shortcoming would be improved by specifying in more detail which IS problems are affected by research results and under what conditions.

There is a critical shortage of empirical studies on IS problems, and the classification presented here is only preliminary. Much more needs to be done in properly classifying and analyzing IS problems. Moreover, we urgently need empirical studies on the frequency of different IS problems, and the IS contexts in which they occur. These investigations should be carried out through case studies and in action research settings because this research strategy seems to be the only means of obtaining sufficiently rich data. In addition, the validity of these research methods is better than that of empirical surveys. IS problems are often confidential, and not easily disclosed as, for example, in questionnaires. Also lacking are empirical studies into the sorts of problems typical of different kinds of information systems. Clearly, an expert system introduces problems different from those in a payroll system. This necessitates the development of a taxonomy of IS applications that would help to relate specific problems to specific types of information systems. The taxonomy should be based on factors other than just characteristics of the operations environment; that is, it should take into account the environments of the information systems and their characteristics. Finally, we need more empirical studies into how proposed measures contribute to IS problem resolution in real-world settings.

Unfortunately, the sort of empirical research strategy suggested here is extremely difficult and time consuming to carry out. The selection of an appropriate research method is often done *de novo* with no prior experience. Moreover, the research method may require longitudinal research designs, which may be extremely difficult to set up and implement. Furthermore, research consumes resources, and the results may be inconclusive owing to various "noise factors." In order to make studies comparable, the researcher has to account for differences in organizational and user environments and analyze their influence on the results obtained. Often this task is far from trivial.

Another question raised by the survey is: How can we improve the practices of information systems, and where is the likely region of success to be? The first observation here is that there is no clear-cut general answer to this question. We do not have the best systems development approach, nor do we have the best IS theory. Every discussed measure and theory is effective within certain limits for certain classes of IS problems. This stresses the importance of the careful analysis of IS contexts, where measures and perspectives are matched with the perceived problems. This, however, requires continual analysis, reflection, and assessment of IS contexts and the creation of appropriate social and organizational conditions to foster these processes. We believe that only through such arrangements can the significance of the measures discussed be perceived. Single measures are seldom valuable if applied blindly; they may prove excellent if they are applied in a critical way to IS contexts.

The problems that have the greatest potential for being solved effectively in the near future are those in technology, process features, and operations. Their effective resolution may also reduce some of the problems relating to people. Many traditional IS applications can now be developed "easily" with capital-intensive technologies. This eases interaction with the users and makes development times shorter and maintenance easier. Also changes in the user environment improve the situation: Users are more knowledgeable, and they know what to expect from information systems. All this improves communication and creates realistic expectations. This applies only to some areas of IS development, however. Large-scale IS, expert systems, and new office applications may exhibit a new range of IS problems. In particular, the growing interest in organization-wide data resource management, expert systems, and competitive applications will amplify problems relating to data, conceptual understanding, people's reactions, and complexity. This is the research challenge for the IS community in the next decade.

## ACKNOWLEDGMENTS

The research was funded in part by the Academy of Finland. The author acknowledges helpful contributions from Heinz Klein and the critical comments of referees, which helped to improve substantially the organization and contents of the paper.

## REFERENCES

ACKOFF, R. L. 1967. Management misinformation systems. *Manage. Sci. 14*, 4, B147–B156.

ADAMS, C. R. 1975. How management users view information systems. *Decis. Sci. 6*, 2, 337–345.

AHITUV, N., AND NEUMANN, S. 1984. A flexible approach to information system development. *MIS Q. 8*, 2, 69–78.

ALTER, S. 1980. *Decision Support Systems: Current Practice and Continuing Challenges.* Addison-Wesley, Reading, Mass.

ALTER, S., AND GINZBERG, M. J. 1978. Managing uncertainty in MIS implementation. *Sloan Manage. Rev. 19*, 23–31.

ANSI/SPARC 1975. ANSI/X3/SPARC study group on data base management systems, interim report. *FDT Bull. ACM SIGMOD 7*, 21.

ANTHONY, R. 1965. *Planning and Control Systems: A Framework Analysis.* Harvard University Press, Boston.

APPLETON, D. S. 1983. Data-driven prototyping. *Datamation 29*, 11, 259–268.

ATTEWELL, P., AND RULE, J. 1984. Computing and organizations: What we know and what we don't know. *Commun. ACM 27*, 12 (Dec.), 1184–1192.

BEATH, C. M. 1983. Strategies for managing MIS projects: A transaction cost approach. In *Proceedings of the 4th International Conference on Information Systems* (Houston, Tex., Dec. 15–17). The Institute of Management Sciences, pp. 133–147.

BELL, T. E., AND THAYER, T. A. 1976. Software requirements: Are they really a problem? In *Proceedings of the 2nd International Conference of Software Engineering* (San Francisco, Oct.). IEEE, New York, pp. 61–68.

BERGER, P. L., AND LUCKMANN, T. 1967. *The Social Construction of Reality.* Penguin Books, Harmondsworth, England.

BERILD, S., BUBENKO, J., JR., AND LINDENCRONA, E. 1984. Conceptual modeling of applications—Theory and practice. SYSLAB Working Paper No. 77, Department of Information Processing and Computer Science, SYSLAB-S, Univ. of Stockholm, Stockholm, Sweden.

BLOOR, D. 1976. *Knowledge and Social Imagery.* Routledge & Kegan Paul, London.

BLUMENTHAL, S. C. 1969. *Management Information Systems: A Framework for Planning and Development.* Prentice-Hall, Englewood Cliffs, N.J.

BLUMER, H. 1969. *Symbolic Interactionism: Perspective and Method.* Prentice-Hall, Englewood Cliffs, N.J.

BOEHM, B. W. 1976. Software engineering. *IEEE Trans. Comput. C-25* (Dec.), 1226–1241.

BOEHM, B. W. 1981. *Software Economics.* M.I.T. Press, Boston.

BOLAND, R. J., JR. 1978. The process and product of system design. *Manage. Sci. 24*, 9, 887–898.

BOLAND, R. J., JR. 1979. Control, causality and information system requirements. *Account. Organ. Soc. 4*, 5, 259–272.

BOLAND, R. J., JR. 1981. A study in system design. C. West Churchman and Chris Argyris. *Account. Organ. Soc. 6*, 2, 109–118.

BOLAND, R. J., JR. 1984. Sense-making of accounting data as a technique of organizational diagnosis. *Manage. Sci. 30*, 7, 868–882.

BOLAND, R. J., JR. 1985. Phenomenology: A preferred approach to research on information systems. In *Research Methods in Information Systems*, E. Mumford, R. Hirschheim, G. Fitzgerald, and A. T. Wood-Harper, Eds. North-Holland, Amsterdam, pp. 193–203.

BOLAND, R. J., JR. AND DAY, W. 1982. The process of system design: A phenomenological approach. In *Proceedings of the 3rd International Conference on Information Systems* (Ann Arbor, Mich., Dec. 13–15), M. Ginzberg and C. Ross, Eds. The Institute of Management Sciences, pp. 31–45.

BOLAND, R. J., JR. AND PONDY, L. R. 1981. Accounting in organizations: A union of natural and rational perspectives. BEPR Faculty Working Paper No. 811, College of Commerce and Business Administration, Univ. of Illinois Bureau of Economic and Business Research, Urbana-Champaign.

BORGIDA, A. 1985. Features of languages for the development of information systems at the conceptual level. *IEEE Software 2*, 1, 63–72.

BOSTROM, R. B. 1978. A MIS metaframework. In *Proceedings of the National Conference on Decision Systems.* Society for Management Information Science, New York, pp. 164–166.

BOSTROM, R. B., AND HEINEN, J. S. 1977a. MIS problems and failures. A socio-technical perspective: Part I: The causes. *MIS Q. 1*, 3, 17–32.

BOSTROM, R. B., AND HEINEN, J. S. 1977b. MIS problems and failures: A socio-technical perspective: Part II: The application of socio-technical theory. *MIS Q. 1*, 4, 11–28.

BOYNTON, A. C., AND ZMUD, R. W. 1984. An assesment of critical success factors. *Sloan Manage. Rev. 26*, 1, 17–27.

BRIEFS, U. 1980. The impact of computerization on the working class and the role of trade unions. In *The Human Side of Information Processing*, N. Bjorn-Andersen, Ed. North-Holland, Amsterdam, pp. 53–68.

BROOKS, S. 1974. *A Mythical Man-Month*. Addison-Wesley, Reading, Mass.

BUBENKO, J. A., JR. 1980. Information modeling in the context of systems development. In *Information Processing 80*, S. Lavington, Ed. North-Holland, Amsterdam.

BUBENKO, J. A., JR. 1983. Information and data modeling: State of the art and research directions. In *Proceedings of the 2nd Scandinavian Research Seminar on Information Modelling and Data Base Management, Acta Universitas Tamperensis, Ser. B, vol. 19*, H. Kangassalo, Ed. Univ. of Tampere, Tampere, Finland, pp. 717–720.

BUBENKO, J. A., JR. 1986. Information system methodologies—A research view. SYSLAB Rep. No. 40, The Systems Development and Artificial Intelligence Laboratory, Univ. of Stockholm, Stockholm, Sweden.

BUCKINGHAM, R. A., Ed. 1977. *Education and Large Information Systems*. North-Holland, Amsterdam.

BURRELL, G., AND MORGAN, G. 1979. *Sociological Paradigms and Organizational Analysis*. Heinemann, London.

CANNING, R. G. 1977. Getting the requirements right. *EDP Anal. 15*, 7, 1–14.

CANNING, R. G. 1983. Plan now for work-stations. *EDP Anal. 21*, 2.

CANNING, R. G. 1984. The changing computing environment. *EDP Anal. 22*, 9.

CARLSON, E. D. 1974. Evaluating the impact of information systems. *Manage. Inf. 3*, 2, 57–67.

CARLSON, W. M. 1979. Business information analysis and integration technique (BIAIT)—The new horizon. *Data Base 10*, 4 (spring), 3–9.

CHAPIN, N. 1979. Some structured analysis techniques. *Data Base 10*, 3, 16–23.

CHECKLAND, P. 1981. *Systems Thinking, Systems Practice*. Wiley, Chichester, England.

CHECKLAND, P. 1984. Systems theory and information systems. In *Beyond Productivity: Information Systems Development for Organizational Effectiveness*. Th. M. A. Bemelmans, Ed. North-Holland, Amsterdam, pp. 9–22.

CHUA, W. F., LAUGHLIN, R. C., LOWE, E. A., AND PUXTY, A. G. 1981. An integrated epistemology of theory and practical action for management science. Paper presented at the Management Epistemology Workshop of the European Institute for Advanced Studies in Management, University of Lancaster, Oct. 19–21.

CHURCHMAN, C. W. 1971. *The Design of Inquiring Systems*. Basic Books, New York.

CIBORRA, C. U. 1981. Information systems and transactions architecture. *J. Policy Anal. Inf. Syst. 5*, 4, 305–324.

CIBORRA, C. U. 1983. The social costs of information technology and participation in system design. In *Systems Design for, with and by the Users*, U. Briefs, C. Ciborra, and L. Schneider, Eds. North-Holland, Amsterdam, pp. 41–50.

CIBORRA, C. U. 1984. Information systems and organizational exchange: A new design approach. In *Beyond Productivity: Information Systems for Organizational Effectiveness*, Th. M. A. Bemelmans, Ed. North-Holland, Amsterdam, pp. 135–145.

CIBORRA, C. U., AND BRACCHI, G. 1983. Systems development and auditing in turbulent contexts: Towards a new participative approach. In *Information Systems Auditing*, E. M. Wysong and I. de Lotto, Eds. North-Holland, Amsterdam, pp. 41–52.

COOPER, R. B., AND SWANSON, E. B. 1979. Management information requirements assessment. The state of the art. *Data Base 11*, 2, 5–16.

COURBON, J. C., AND BOURGEOIS, M. 1980. The information systems designer as a nurturing agent of a socio-technical process. In *The Information Systems Environment*, H. C. Lucas, F. F. Land, T. J. Lincoln, and K. Supper, Eds. North-Holland, Amsterdam, pp. 139–148.

CRON, W. L., AND SOBOL, M. G. 1983. The relationship between computerization and performance: A strategy for maximizing economic benefits of computerization. *Inf. Manage. 6*, 3, 171–182.

DAVIS, G. B. 1982. Strategies for information requirements determination. *IBM Syst. J. 21*, 1, 4–30.

DAVIS, G. B. 1984. Management of the information systems resource: Introduction. In *The Inforation Systems Research Challenge*, F. W. McFarlan, Ed. Harvard Business School Press, Boston, Mass., pp. 191–195.

DAVIS, G. B., AND OLSON, M. 1985. *Management Information Systems—Conceptual Foundations Methods and Development* (2nd ed.). McGraw-Hill, New York.

DEBRABANDER, B., AND THIERS, G. 1984. Successful information system development in relation to situational factors which affect effective communication between MIS-users and EDP-specialists. *Manage. Sci. 30*, 2, 137–155.

DE CINDIO, F., DE MICHELIS, G., SIMONE, C., VASSALLO, R., AND ZANABONI, A. M. 1986. CHAOS as coordination technology: Conversation handling active office systems. Working Paper, Dipart Science dell'Informazione, Univ. di Milano, Milano, Italy.

DEMAIO, A. 1980. Socio-technical methods for information systems design. In *The Information Systems Environment*, H. Lucas, F. Land, T. Lincoln, and K. Supper, Eds. North-Holland, Amsterdam, pp. 105–122.

DEMARCO, T. 1978. *Structured Analysis and System Specification.* Yourdon Press, New York.

DEMOS 1979. Systems development by negotiations: A tentative model. In *Computers Dividing Man and Work*, A. Sandberg, Ed. Arbetslivscentrum, Malmö, Sweden, pp. 129–146.

DICKSON, G. W., SENN, J. A., AND CHERVANY, N. L. 1977. Research in management information systems: The Minnesota experiments. *Manage. Sci. 23*, 9, 913–923.

DICKSON, G. W., LEITHEISER, R. L., WETHERBE, J. C., AND NECHIS, M. 1984. Key information systems issues for the 1980s. *MIS Q. 8*, 3, 135–159.

EARL, M. J., AND HOPWOOD, A. G. 1980. From management information to information management. In *The Information Systems Environment*, H. Lucas, F. Land, T. Lincoln, and K. Supper, Eds. North-Holland, Amsterdam, pp. 3–12.

EHN, P., AND SANDBERG, A. 1979. Systems development: Critique of ideology and the division of labor in computer field. In *Computer Dividing Man and Work*, A. Sandberg, Ed. Arbetslivscentrum, Malmö, Sweden, pp. 34–46.

EIN-DOR, P., AND SEGEV, E. 1978. Organizational context and the success of management information systems. *Manage. Sci. 24*, 10, 1064–1077.

EMERY, J. C. 1971. Cost/benefit analysis of information systems. Lecture notes prepared for SMIS Workshop on Cost/Benefit Analysis, Wharton School, Univ. of Pennsylvania, Pittsburgh.

EMERY, J. C. 1974. Costs and benefits of information systems. In *Information Processing 74*. North-Holland, Amsterdam, pp. 967–971.

EMERY, J. C. 1977. Managerial and economic issues in distributed computing. In *Information Processing 77*, B. Gilchrist, Ed. North-Holland, Amsterdam, pp. 945–955.

EMERY, F. E. 1972. Characteristics of socio-technical systems. In *Design of Jobs*, L. E. Davis and J. C. Taylor, Eds. Penguin Books, Harmondsworth, England, pp. 177–198.

EPISKOPOU, D. M., AND WOOD-HARPER, A. T. 1984. A multi-view methodology—Applications and implications. In *Beyond Productivity: Information Systems Development for Organizational Effectiveness*, Th. M. A. Bemelmans, Ed. North-Holland, Amsterdam, pp. 189–211.

FLAVIN, M. 1981. *Fundamental Concepts of Information Modeling.* Yourdon Press, New York.

FLORES, F., AND LUDLOW, J. J. 1981. Doing and speaking in the office. In *Decision Support Systems—Issues and Challenges*, G. Fick and R. H. Sprague, Eds. Pergamon, London, pp. 95–118.

FLOYD, C. 1986. A comparative evaluation of system development methods. In *Improving the Systems Development Practice (CRIS-III)*, B. Olle, H. G. Sol, and A. A. Verrijn-Stuart, Eds. North-Holland, Amsterdam, pp. 19–54.

FRANZ, C. R., AND ROBEY, D. 1984. An investigation of user-led system design: Rational and political perspectives. *Commun. ACM 27*, 12 (Dec.), 1202–1209.

FURTADO, A. L. 1983. An informal approach to formal specifications. *ACM SIGMOD Rec. 8*, 1, 32–41.

GALBRAITH, J. 1977. *Organization Design.* Addison-Wesley, Reading, Mass.

GARFINKEL, H. 1967. *Studies in Ethnomethodology.* Prentice-Hall, Englewood Cliffs, N.J.

GINZBERG, M. J. 1975. A process approach to management science implementation. Ph.D. dissertation, Alfred P. Sloan School of Management, Massachusetts Institute of Technology, Boston.

GINZBERG, M. J. 1978. Steps towards more effective implementation of MS and MIS. *Interfaces 8*, 3, 295–300.

GINZBERG, M. J. 1980. An organizational contingencies view of accounting and information systems implementation. *Account. Org. Soc. 5*, 4, 369–382.

GINZBERG, M. J. 1981. Early diagnosis of MIS implementation failure: Promising results and unanswered questions. *Manage. Sci. 27*, 4, 459–478.

GLADDEN, G. R. 1982. Stop the life-cycle, I want to get off. *ACM SIGSOFT Softw. Eng. Notes 7*, 2, 35–39.

GOLDKUHL, G. 1980. *Människor, Information och Kommunikation—en Humaninfologisk Referensram för Informationsystem.* Institutionen för Administrativ Databehandling, Univ. of Stockholm, Stockholm, Sweden (in Swedish).

GOLDKUHL, G., AND LYYTINEN, K. 1982. A language action view of information systems. In *Proceedings of the 3rd International Conference on Information Systems* (Ann Arbor, Mich., Dec. 13–15), M. Ginzberg and C. Ross, Eds. The Institute of Management Sciences, pp. 13–30.

GOLDKUHL, G., AND LYYTINEN, K. 1984. Information system specification as rule reconstruction. In *Beyond Productivity: Information Systems for Organizational Effectiveness*, Th. M. A. Bemelmans, Ed. North-Holland, Amsterdam, pp. 30–55.

GÖRANZÖN, B., Ed. 1983. *Datautvecklingens filosofi: Tyst kunskap och ny teknik.* Carlsson & Jonsson, Stockholm, Sweden (in Swedish).

GUPTA, A. 1982. Emerging trends in office technology. In *Office Information Systems*, N. Naffaf, Ed. North-Holland, Amsterdam, pp. 599–626.

GUSTAFSSON, M. R., KARLSSON, T., AND BUBENKO, J. A. 1982. A declarative approach to conceptual information modelling. SYSLAB Tech. Rep. 8, Dept. of Information Processing and Computer Science, Univ. of Stockholm, Stockholm, Sweden.

HABERMAS, J. 1979. *Communication and the Evolution of Society* (transl. T. McCarthy). Heinemann, London.

HAMILTON, S. 1980. Evaluation of information system effectiveness: A comparison of evaluation approaches and evaluation viewpoints. MISRC-WP-81-03, MISRC, Graduate School of Business Administration, Univ. of Minnesota, Minneapolis.

HAMMER, M. 1981. Life cycle management. *Inf. Manage. 4*, 4, 71–80.

HEDBERG, B. 1980. Using computerized information systems to design better organizations and jobs. In *The Human Side of Information Processing*, N. Bjorn-Anderssen, Ed. North-Holland, Amsterdam, pp. 19–33.

HEDBERG, B., AND JÖNSSÖN, S. 1978. Designing semi-confusing information systems for organizations in changing environments. *Account. Organ. Soc. 3*, 1, 47–64.

HIRSCHHEIM, R. A. 1983. Assessing participative system design: Some conclusions from an exploratory study, *Inf. Manage. 6*, 4, 317–327.

HIRSCHHEIM, R. A. 1985. User experience with and assessment of participative systems design. *MIS Q. 9*, 4, 295–303.

HIRSCHHEIM, R. A. 1986. The effect of a priori views on the social implications of computing: The case of office automation. *ACM Comput. Surv. 18*, 2 (June), 165–195.

HIRSCHHEIM, R. A., AND SMITHSON, S. 1987. A critical analysis of information systems evaluation. In *Information Systems Assessment*, G. B. Davis and N.-B. Andersen, Eds. North-Holland, Amsterdam (to appear).

HIRSCHHEIM, R. A., LAND, F., AND SMITHSON, S. 1984. Implementing computer-based information systems in organizations—Issues and strategies. In *Proceedings of the Interact-84* (London, Sept. 4–6). North Holland, Amsterdam, pp. 356–369.

HOROWITZ, E., KEMPER, A., AND NARASIMHAN, B. 1985. Application generators. *IEEE Software 2*, 1, 40–54.

HØYER, R. 1980. User participation: Why is development so slow? In *The Information Systems Environment*, H. C. Lucas, F. F. Land, T. J. Lincoln, and K. Supper, Eds. North-Holland, Amsterdam, pp. 129–138.

HUBER, G. P. 1981. Organizational science contributions to the design of decision support systems. In *Decision Support Systems: Issues and Challenges*, G. Fick and R. H. Sprague, Eds. Pergamon, London, pp. 45–55.

HUBER, G. P. 1983. Cognitive style as a basis for MIS and DSS designs: Much ado about nothing? *Manage. Sci. 29*, 5, 567–577.

IBM 1981. Business systems planning—Information systems planning guide. Application manual, GE20-0527-3 (3rd ed.), IBM Corp.

IIVARI, J. 1983. Contributions to the theoretical foundations of systemeering research and the PIOCO-model, Ph.D. dissertation Acta Universitatis Ouluensis, Ser. A 150, Institute of Data Processing, Univ. of Oulu, Oulu, Finland.

IVES, B., AND LEARMONTH, G. P. 1984. Information system as a competitive weapon. *Commun. ACM 27*, 12 (Dec.), 1193–1201.

IVES, B., AND OLSON, M. 1981. User involvement in information systems: A critical review of the empirical literature. CRIS 15, GBA 81-07, Center for Research on Information Systems, Graduate School of Business Administration, New York Univ., New York.

IVES, B., HAMILTON, S., AND DAVIS, G. 1980. A framework for research in computer based management information systems. *Manage. Sci. 26*, 9, 910–934.

JACKSON, M. 1983. *Systems Development.* Prentice-Hall, Englewood Cliffs, N.J.

JARDINE, D. A., AND REUBER, A. R. 1984. Information semantics and the conceptual schema. *Inf. Syst. 9*, 2, 147–156.

JENKINS, A. M. 1983. Prototyping: A methodology for the design and development of application systems. Discussion Paper No. 227, Indiana Univ. Graduate School of Business, Division of Research, Bloomington.

JUNG, C. G. 1931. Psychological types. In *The Basic Writings of C. G. Jung.* Modern Library, New York, pp. 183–285.

KAHN, B. K. 1982. LDDM—A structured logical database design methodology. In *Data Base Design Techniques I: Requirements and Logical Structures*, S. B. Yao, S. B. Navathe, J. L. Weldon, and T. L. Kunii, Eds. Springer-Verlag, Berlin, pp. 31–55.

KAY, R. H., SZYPERSKI, N., HORING, K., AND BARTZ, G. 1981. Stratefic planning of information systems at the corporate level. *Inf. Manage. 3*, 2, 175–186.

KEEN, P. G. W. 1974. Toward behavioral methodology for the study of OR/MS implementation. Sloan Working Paper No. 701-74, Sloan School of Management, Massachusetts Institute of Technology, Boston, Mass.

KEEN, P. G. W. 1981a. Decision support systems: A research perspective. In *Decision Support Systems: Issues and Challenges*, G. Fick and R. L. Sprague, Jr., Eds. Pergamon, Oxford, England, pp. 23–44.

KEEN, P. G. W. 1981b. Information systems and organizational change. *Commun. ACM 24*, 1 (Jan.), 24–33.

KEEN, P. G. W., AND SCOTT MORTON, M. S. 1978. *Decision Support Systems: An Organizational Perspective.* Addison-Wesley, Reading, Mass.

KENSING, F. 1984. Towards evaluation of methods for property determination—A framework and critique of Yordon-DeMarco approach. In *Beyond Productivity: Information Systems Development for Organizational Effectiveness*, Th. M. A. Bemelmans, Ed. North-Holland, Amsterdam, pp. 325–338.

KENT, W. 1978. *Data and Reality.* North-Holland, Amsterdam.

KEROLA, P. 1980. On infological research into the systemeering process. In *The Information Systems Environment*, H. Lucas, F. Land, T. Lincoln, and K. Supper, Eds. North-Holland, Amsterdam, pp. 187–198.

KEROLA, P., AND FREEMAN, P. 1981. A comparison of lifecycle models. In *Proceedings of the 5th International Conference on Software Engineering* (San Diego, Calif., Mar.). IEEE, New York, pp. 1–10.

KIMBROUGH, S. O., LEE, R. M., AND NESS, D. 1984. Performative, informative and emotive systems: The first piece of the PIE. In *Proceedings of the 5th International Conference on Information Systems* (Tucson, Ariz., Nov. 28–30). Society for Management Information Systems, New York, pp. 28–30.

KING, J. L. 1983. Centralized versus decentralized computing: Organizational considerations and management options. *ACM Comput. Surv. 15*, 4 (Dec.), 319–349.

KING, J. L., AND KRAEMER, K. L. 1984. Evolution and organizational information systems: An assessment of Nolan's stage model. *Commun. ACM 27*, 5 (May), 466–475.

KING, J. L., AND SCHREMS, E. L. 1978. Cost–benefit analysis in information systems development and operation. *ACM Comput. Surv. 10*, 1 (Mar.), 19–34.

KING, W. R. 1978. Strategic planning for management information systems. *MIS Q. 2*, 1, 27–37.

KLEIJNEN, J. P. C. 1980. *Computers and Profits—Quantifying Financial Benefits of Information System.* Prentice-Hall, Englewood Cliffs, N.J.

KLEIN, H. K., AND HIRSCHHEIM, R. A. 1983. Issues and approaches in appraising technological change in the office: A consequentalist perspective. *Office Technol. People 2*, 1, 15–42.

KLEIN, H. K., AND HIRSCHHEIM, R. A. 1987. Social change and the future of information systems development. In *Critical Issues in Information Systems Research*, R. J. Boland and R. A. Hirschheim, Eds. Wiley, Chichester, England (to appear).

KLEIN, H. K., AND WELKE, R. J. 1980. An evaluation of the Finnish PSC-systemeering (systems development) theory. In *Report of the 3rd Scandinavian Research Seminar on Systemeering Models*, K. Lyytinen and E. Peltola, Eds. Univ. of Jyväskylä, Department of Computer Science, Jyväskylä, Finland, pp. 2–37.

KLING, R. 1980. Social analyses of computing: Theoretical perspectives in recent empirical research. *ACM Comput. Surv. 12*, 1 (Mar.), 61–110.

KLING, R., AND IACONO, S. 1984. The control of information systems developments after implementation. *Commun. ACM 27*, 12 (Dec.), 1218–1226.

KLING, R., AND SCACCHI, W. 1982. The social web of computing: Computer technology as social organization. *Advan. Comput. 21*, 2–90.

KOLB, D. A., AND FROHMAN, A. L. 1970. An organization development approach to consulting. *Sloan Manage. Rev. 11*, 51–65.

KUBICEK, H. 1983. User participation in system design: Some questions about the structure and content arising from recent research from a trade union perspective. In *Systems Design for, with, and by the Users*, U. Briefs, C. Ciborra, and L. Schneider, Eds. North-Holland, Amsterdam, pp. 3–18.

KUMAR, K., AND WELKE, R. J. 1984. Implementation failure and system developer values. In *Proceedings of the 5th International Conference on Information Systems* (Tucson, Ariz., Nov. 20–30), L. Maggi, J. L. King, and K. L. Kraemer, Eds. Society for Management Information Systems, New York, pp. 1–14.

KUNG, C. H., AND SØLVBERG, A. 1984. An exercise in integrating database design tools. In *Report of the 3rd Scandinavian Research Seminar on Information Modelling and Data Base Management*, H. Kangassalo, Ed. Univ. of Tampere, Tampere, Finland, pp. 277–298.

LAND, F. 1982. Adapting to changing user requirements. *Inf. Manage. 5*, 1, 59–75.

LAND, F., MUMFORD, E., AND HAWGOOD, J. 1980. Training the systems analyst of the 1980s: Four analytical procedures to assist the design process. In *The Information Systems Environment*, H. Lucas, F. Land, T. Lincoln, and K. Supper, Eds. North-Holland, Amsterdam, pp. 239–256.

LANDRY, M., AND LE MOIGNE, J.-L. 1977. Towards a theory of organizational information system—A general system perspective. In *Information Processing 77*, B. Gilchrist, Ed. North-Holland, Amsterdam, pp. 801–805.

LANGEFORS, B. 1973. *Theoretical Analysis of Information Systems.* Studentlitteratur, Lund, Sweden.

LANZARA, G. F. 1983. The design process: Frames, metaphors and games. In *Systems Design for, with, and by the Users*, U. Briefs, C. Ciborra, and L. Schneider, Eds. North-Holland, Amsterdam, pp. 29–40.

LANZARA, G. F., AND MATHIASSEN, L. 1985. Mapping situations within a system development project. *Inf. Manage. 8*, 1, 3–20.

LEDGARD, H., SINGER, A., AND WHITESIDE, J. 1981. *Directions in Human Factors for Interactive Systems.* Springer-Verlag, Berlin.

LEHMAN, M. M. 1980. Programs, life cycles, and laws of software evolution. *Proc. IEEE 68*, 1060–1076.

LEHMAN, M. M. 1984. Program evolution. *Inf. Process. Manage. 20*, 1–2, 19–36.

LEHTINEN, E., AND LYYTINEN, K. 1986. Action based model of information system. *Inf. Syst. 11*, 4, 299–317.

LEWIN, K. 1952. Group decision and social change. In *Readings in Social Psychology*, T. M. Newcomb and E. L. Hartley, Eds. Holt, New York, pp. 330–344.

LUCAS, H. C., JR. 1975. *Why Information Systems Fail.* Columbia University Press, New York.

LUCAS, H. C., JR. 1978. The evolution of an information system: From key-man to every person. *Sloan Manage. Rev. 19,* 2, 39–52.

LUCAS, H. C., JR.. 1981. *Implementation: The Key to Successful Information Systems.* Columbia University Press, New York.

LUNDEBERG, M., GOLDKUHL, G., AND NILSSON, A. 1981. *Information Systems Development—A Systematic Approach.* Prentice-Hall, Englewood Cliffs, N.J.

LYYTINEN, K. 1985. Implications of theories of language for information systems. *MIS Q. 9,* 1, 61–74.

LYYTINEN, K. 1987a. Two views of information modeling. *Inf. Manage. 11* (to appear).

LYYTINEN, K. 1987b. Information system failure: An exploratory survey of analysts' perceptions. Working Paper, Dept. of Computer Science, Univ. of Jyväskylä, Jyväskylä, Finland.

LYYTINEN, K., AND HIRSCHHEIM, R. A. 1987. Information system failures: A survey and classification of empirical literature. *Oxford Surv. Inf. Technol. 4* (to appear).

LYYTINEN, K., AND LEHTINEN, E. 1984a. On information modeling through illocutionary logic. In *Report of the 3rd Scandinavian Research Seminar on Information Modeling and Data Base Management,* H. Kangassalo, Ed. Univ. of Tampere, Tampere, Finland, pp. 35–118.

LYYTINEN, K., AND LEHTINEN, E. 1984b. Discourse analysis as an information system specification method. In *Report of the 7th Scandinavian Research Seminar on Systemeering* (Part I). Helsinki Business School, Helsinki, Finland, pp. 146–200.

MARCH, J. G., AND SIMON, H. A. 1958. *Organizations.* Wiley, New York.

MARKUS, M. L. 1983. Power, politics and MIS implementation. *Commun. ACM 26,* 6 (June), 430–444.

MARKUS, M. L., AND PFEFFER, J. 1983. Power and the design and implementation of accounting and control systems. *Account. Organ. Soc. 8,* 2/3, 205–218.

MARKUS, M. L., AND ROBEY, D. 1983. The organizational validity of management information systems. *Human Relat. 36,* 203–226.

MARTIN, J. 1982. *Application Development without Programmers.* Prentice-Hall, Englewood Cliffs, N.J.

MARTIN, J. 1983. *Managing the Data-Base Environment.* Prentice-Hall, Englewood Cliffs, N.J.

MARTIN, J. 1985. Computer science education today: A dialogue. *Commun. ACM 28,* 3 (Mar.), 251–255.

MARX, K. 1976. *Capital: a Critique of Political Economy* (Transl. B. Fowkes), I–III. Penguin Books, Harmondsworth, England. (Vol. I first published 1867, vol. II 1885, vol. III 1894.)

MASON, R. O., AND MITROFF, I. I. 1973. A program for research on management information systems. *Manage. Sci. 19,* 5, 475–487.

MASON, R. O., AND MITROFF, I. I. 1981. *Challenging Strategic Planning Assumptions.* Wiley, New York.

MATHIASSEN, L., AND ANDERSEN, P. B. 1983. Nurses and semiotics: The impact of EDP-based systems upon professional languages. In *Report of the 6th Scandinavian Research Seminar on Systemeering,* H. Nurminen and T. Gaupholm, Eds. Institute of Information Science, Univ. of Bergen, Bergen, Norway, pp. 227–262.

MAYNTZ, R. 1984. Information systems and organizational effectiveness—A sociological perspective. In *Beyond productivity: Information systems development for organizational effectiveness,* Th. M. A. Bemelmans, Ed. North-Holland, Amsterdam, pp. 173–188.

MCCRACKEN, D. D., AND JACKSON, M. A. 1981. A minority dissenting position. In *Systems Analysis and Design: A Foundation for the 1980s,* W. W. Cotterman, J. D. Couger, N. L. Enger, and F. Harold, Eds. North-Holland, Amsterdam, pp. 551–553.

MITROFF, I. I. 1980. Toward a logic and methodology for real-world problems. In *The Human Side of Information Processing,* N. Bjorn-Andersen, Ed. North-Holland, Amsterdam, pp. 187–195.

MITROFF, I. I. 1982. Archetypal social systems analysis: On the deeper structure of human systems. Graduate School of Business, Univ. of Southern California, Los Angeles.

MITROFF, I. I., AND MASON, R. O. 1983. Can we design systems for managing messes? or why so many management information systems are uninformative. *Account. Organ. Soc. 8,* 2/3, 195–203.

MORAN, T. P. 1981. An applied psychology of the user. *ACM Comput. Surv. 13,* 1 (Mar.), 1–11.

MORGAN, H., AND SODEN, J. 1973. Understanding MIS failures. *Data Base 5,* 1.

MUMFORD, E. 1981. Participative systems design: Structure and method. *Syst. Objectives Solutions 1,* 1, 5–19.

MUMFORD, E. 1983. *Designing Human Systems—The ETHICS Method.* Manchester Business School, Cheshire, England.

MUMFORD, E. 1984. Participation—From Aristotle to Today. In *Beyond Productivity: Information Systems Development for Organizational Effectiveness,* Th. M. A. Bemelmans, Ed. North-Holland, Amsterdam, pp. 95–104.

MUMFORD, E., HIRSCHHEIM, R. A., FITZGERALD, G., AND WOOD-HARPER, A. T., Eds. 1985. *Research Methods in Information Systems.* North-Holland, Amsterdam.

NEWELL, A., AND SIMON, H. A. 1972. *Human Problem Solving.* Prentice-Hall, Englewood Cliffs, N.J.

NOLAN, R. L. 1973. Managing the computer resource: A stage hypothesis. *Commun. ACM 16*, 7 (July), 399–405.

NOLAN, R. L. 1977. *Management Accounting and Control of Data Processing*. National Association of Accountants, New York.

NOLAN, R. L. 1979. Managing the crisis in data processing. *Harvard Bus. Rev. 57*, 3, 115–126.

NOLAN, R. L. 1984. Managing the advanced stages of computer technology: Key research issues. In *The Information Systems Research Challenge*, F. W. McFarlan, Ed. Harvard University Press, Boston, Mass., pp. 195–215.

NOLAN, R. L., AND WETHERBE, J. C. 1980. Towards a comprehensive framework for MIS research. *MIS Q. 4*, 2, 1–20.

NUNAMAKER, J. F., AND KONNSYNSKI, B. 1981. Formal and automated techniques of systems analysis and design. In *Systems Analysis and Design—A Foundation for the 1980's*, W. W. Cotterman, J. D. Couger, N. L. Enger, and F. Harold, Eds. North-Holland, Amsterdam, pp. 291–320.

NYGAARD, K. 1983. Partition in systems development—The tasks ahead. In *Systems Design for, with and by the Users*. U. Briefs, C. Ciborra, and L. Schneider, Eds. North-Holland, Amsterdam, pp. 15–25.

NYGAARD, K., AND HANDLYKKEN, P. 1981. The systems development process—Its setting, some problems and needs for methods. In *Software Engineering Environment*, H. Hunke, Ed. North-Holland, Amsterdam, pp. 157–172.

OLERUP, A. 1980. On a contingency framework of computerized information systems. In *The Information Systems Environment*, H. C. Lucas, F. F. Land, T. J. Lincoln, and K. Supper, Eds. North-Holland, Amsterdam, pp. 83–94.

OLLE, T. W., SOL, H. G., AND VERRIJN-STUART, A. A., Eds. 1982. *Information Systems Design Methodologies: A Comparative Review*. North-Holland, Amsterdam.

OLLE, T. W., SOL, H. G., AND VERRIJN-STUART, A. A., Eds. 1983. *Information Systems Design Methodologies: A Feature Analysis*. North-Holland, Amsterdam.

OUCHI, W. G. 1978. Coupled versus uncoupled control in organizational hierarchies. In *Environments and Organizations*, M. W. Meyer, Ed. Jossey/Bass, San Francisco, Calif.

OUCHI, W. G. 1979. A conceptual framework for the design of organizational control mechanisms. *Manage. Sci. 25*, 9, 833–848.

OUCHI, W. G. 1980. Markets, bureaucracies and clans. *Admin. Sci. Q. 25*, 129–141.

PARNAS, D. L., AND CLEMENTS, P. C. 1985. A rational design process: How and why fake it. In *Proceedings of the International Joint Conference on Theory and Practice of Software Development*. Springer-Verlag, Berlin.

PEARSON, S. 1977. Measurement of computer user satisfaction. Ph.D. dissertation. Arizona State Univ., Tempe.

PETERSON, J. L. 1977. Petri nets. *ACM Comput. Surv. 9*, 3 (Sept.), 223–259.

POWERS, R. F., AND DICKSON, G. W. 1973. MIS project management: Myths, opinions and reality. *California Manage. Rev. 15*, 3, 147–156.

ROBEY, D. 1984. Conflict models for implementation research. In *Management Science Implementation*, R. L. Schultz, Ed. American Elsevier, New York.

ROBEY, D., AND MARKUS, M. L. 1984. Rituals in system design. *MIS Q. 8*, 1, 5–15.

ROCHFELD, A., AND TARDIEU, H. 1983. MERISE: An information system design and development methodology. *Inf. Manage. 6*, 143–159.

ROCKART, J. F. 1979. Critical success factors. *Harvard Bus. Rev. 57*, 2, 81–91.

ROCKART, J. F., AND CRESCENZI, A. D. 1984. Engaging top management in information technology. *Sloan Manage. Rev. 26*, 1, 3–16.

ROSENBERG, R. L. 1980. Incomprehensible computer systems: Knowledge without wisdom. Master's thesis, Laboratory of Computer Science, Massachusetts Institute of Technology, Cambridge, Mass.

ROSENHEAD, J., AND THUNHURST, C. 1982. A materialist analysis of operational research. *J. Oper. Res. Soc. 33*, 1, 111–122.

ROSS, D. T. 1977. Structured analysis (SA): A language for communicating ideas. *IEEE Trans. Softw. Eng. 3*, 1, 16–34.

ROSS, D. T. 1985. Applications and extensions of SADT. *Computer 3*, 4, 25–34.

ROSS, D. T., AND SCHOMAN, K. E., JR. 1977. Structured analysis for requirements definition. *IEEE Trans. Softw. Eng. 3*, 1, 6–15.

SANDBERG, A., Ed. 1979. *Computers Dividing Man and Work*. Arbetslivscentrum, Malmö, Sweden.

SANDBERG, A. 1985. Socio-technical design, trade-union strategies and action research. In *Research Methods in Information Systems*, E. Mumford, R. A. Hirschheim, G. Fitzgerald, and A. T. Wood-Harper, Eds. North-Holland, Amsterdam, pp. 79–92.

SCACCHI, W. 1984. Managing software engineering projects: A social analysis. *IEEE Trans. Softw. Eng. 10*, 1, 49–59.

SCHNEIDER, L., AND CIBORRA, C. 1983. Technology bargaining in Norway. In *Systems Design for, with, and by the Users*, U. Briefs, C. Ciborra, and L. Schneider, Eds. North-Holland, Amsterdam, pp. 243–250.

SEARLE, J. R. 1969. *Speech Acts—An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, England.

SEARLE, J. R. 1979. *Expression and Meaning*. Cambridge University Press, Cambridge, England.

SEN-83, Letter from the Editor. 1983. Computers and human safety. *Softw. Eng. Notes 8*, 5, 1–9.

SENKO, M. 1975. Information systems, records, relations, sets, entities, and things. *Inf. Syst. 1*, 1, 3–13.

SENKO, M. 1977. Conceptual schemas, abstract data structures, enterprise descriptions. In *International Computing Conference 77*, H. Morlet and J. Ribbens, Eds. North-Holland, Amsterdam, pp. 82–102.

SENN, J. A. 1978. A management view of systems analysts: Failures and shortcomings. *MIS Q. 2*, 3, 25–32.

SENN, J. A. 1979. Enhancing the user interface: A research problem in computer-based information systems. WP 79-01, School of Management, State Univ. of New York at Binghamton.

SENN, J. A. 1980. Management's assessment of computer information systems. *J. Syst. Manage. 11*, 9, 6–11.

SHNEIDERMAN, B. 1981. *Software Psychology—Human Factors in Computer and Information Systems*. Winthrop, Cambridge, Mass.

SLUIZER, S., AND CASHMAN, P. M. 1984. XCP: An experimental tool for supporting office procedures. In *Proceedings of the 1st International Conference on Office Automation* (Silver Spring, Md., Oct.). IEEE, New York, pp. 73–80.

SORENSEN, R. E., AND ZAND, D. E. 1975. Improving the implementation of OR/MS models by applying the Lewin–Schein theory of change. In *Implementing Operations Research/Management Science*, R. L. Schultz and D. P. Slevin, Eds. American Elsevier, New York, pp. 217–236.

STACHOWITZ, R. A. 1985. A formal framework for describing and classifying semantic data models. *Inf. Syst. 10*, 1, 77–96.

STAMPER, R. 1984. A logic of social norms and individual affordances: A proposal arising from an investigation in the field of computer and law. Working Paper, London School of Economics, Univ. of London, London.

STAMPER, R. 1985. Management epistemology: Garbage in, garbage out (and what about deontology and axiology). In *Knowledge Representation for Decision Support Systems*, L. B. Methlie and R. H. Sprague, Eds. North-Holland, Amsterdam, pp. 55–77.

STAMPER, R. 1986. A logic of social norms for the semantics of business information. In *Database Semantics (DS-1)*, T. B. Steel and R. Meersman, Eds. North-Holland, Amsterdam, pp. 233–253.

SWANSON, B. E. 1984. Organizational designs for software maintenance. In *Proceedings of the 5th International Conference on Information Systems*, (Tucson, Az., Nov. 28–30), L. Maggi, J. L. King, and K. L. Kraemer, Eds. Society for Management Information Systems, New York, pp. 73–82.

SULLIVAN, C. H. 1985. Systems planning in the information age. *Sloan Manage. Rev. 27*, 1, 3–11.

TAGGART, W. M., AND THARP, M. O. 1977. A survey of information requirements analysis techniques. *ACM Comput. Surv. 9*, 4 (Dec.), 273–290.

TEICHROEW, D., AND HERSHEY, E. A. 1977. PSL/PSA: A computer aided technique for structured documentation and analysis of information processing systems. *IEEE Trans. Softw. Eng. 3*, 1, 48–58.

TEICHROEW, D., MACASOVIC, P., HERSHEY, E. A., III, AND YAMAMOTO, Y. 1980. Application of the entity-relationship approach to information processing systems modeling. In *Entity-Relationship Approach to Systems Analysis and Design*, P. P. Chen, Ed. North-Holland, Amsterdam, pp. 15–36.

THORESEN, K. 1984. Systemutvikling—Produkt og process. Rapport Nr. 747, Norsk Regnecentral, Blindern, Oslo, Norway (in Norwegian).

TRIST, E. L. 1981. The sociotechnical perspective. In *Perspectives on Organization Design and Behavior*, A. H. van de Ven and W. F. Joyce, Eds. Wiley, New York, pp. 19–75.

TSICHRITZIS, D., AND LOCHOVSKY, F. 1982. *Data Models*. Prentice-Hall, Englewood Cliffs, N.J.

TURNER, J. A. 1982. Observations on the use of behavioral models in information systems research and practice. *Inf. Manage. 5*, 3, 207–213.

ULRICH, W. 1981. A critique of pure cybernetic reason: The Chilean experience with cybernetics. *J. Appl. Syst. Anal. 8*, 33–59.

ULRICH, W. 1983. *Critical Heuristics of Social Planning: A New Approach to Practical Philosophy*. Verlag Haupt, Bern, Switzerland.

VAN GRIETHUYSEN, J. J., Ed. 1982. Concepts and terminology for the conceptual schema and the information base. ISO/TC97/SC5/WG3, ISO/TC97 Computers and Information Processing, New York.

VERRIJN-STUART, A. 1979. The past and future of information systems. In *Euro IFIP 79*. North-Holland, Amsterdam, pp. 153–161.

VITALARI, N. P. 1984. Critical assessment of structured analysis methods: A psychological perspective. In *Beyond Productivity: Information Systems Development for Organizational Effectiveness*, Th. M. A. Bemelmans, Ed. North-Holland, Amsterdam, pp. 421–434.

WASSERMAN, A. I. 1980. Software tools and the user software engineering project. Tech. Rep. 46, Laboratory of Medical Information Science, Univ. of California, San Francisco.

WASSERMAN, A. I. 1982. The user software engineering methodology: An overview. In *Information Systems Design Methodologies: A Comparative Review*, T. Olle, H. Sol, and A. Verrijn-Stuart, Eds. North-Holland, Amsterdam, pp. 591–628.

WASSERMAN, A. I., AND FREEMAN, P. 1983. ADA methodologies: Concepts and requirements. *Softw. Eng. Notes 8*, 1, 33–50, ACM SIGSOFT.

WATERS, S. J. 1974. Computer-aided methodology of computer systems design. *Comput. J. 17*, 3, 211–215.

WEGNER, P. 1982. Reflections on capital-intensive software technology. *Softw. Eng. Notes 7*, 4, 24–33.

WEICK, K. E. 1984. Theoretical assumptions and research methodology selection. In *The Information Systems Research Challenge*, F. W. McFarlan, Ed. Harvard University Press, Boston, Mass., pp. 111–132.

WELKE, R. J. 1978. Information system effectiveness evaluation (ISEE). ISRAM PP-7806-1.1, Faculty of Business, McMaster Univ., Hamilton, Ontario, Canada.

WELKE, R. J. 1981. IS/DSS: DBMS support for information systems development. ISRAM WP-8105-1.0, Faculty of Business, McMaster Univ., Hamilton, Ontario, Canada.

WIEDERHOLD, G. 1977. *Database Design*, McGraw-Hill, Tokyo.

WILLIAMSON, O. E. 1975. *Markets and Hierarchies: Analysis and Antitrust Implications*. Free Press, New York.

WILLIAMSON, O. E. 1979. Transaction-cost economics: The governance of contractual relations. *J. Law Econ. 22*, 2, 233–261.

WILLIAMSON, O. E., AND OUCHI, W. G. 1981. The markets and hierarchies and visible hand perspectives. In *Perspectives on Organization Design and Behavior*, A. H. van de Ven and W. F. Joyce, Eds. Wiley, New York, pp. 347–370.

WINOGRAD, T., AND FLORES, F. 1986. *Understanding Computers and Cognition: A New Foundation for Design*. Ablex, Norwood, N.J.

WITTGENSTEIN, L. 1958. *Philosophical Investigations*. Basil Blackwell, Oxford, England.

WÜNDERLICH, D. 1979. *Foundations of Linguistics*. Cambridge University Press, Cambridge, England. (German original 1974.)

WUTHNOW, R., HUNTER, J. D., BERGESEN, A., AND KURZWEIL, E. 1984. *Cultural Analysis*. Routledge & Kegan Paul, Boston, Mass.

YADAV, S. B. 1983. Determining an organization's information requirements: A state of the art survey. *Data Base 14*, 2, 3–20.

YAO, S. B., NAVATHE, S. B., AND WELDON, J.-L. 1982. An integrated approach to database design. In *Data Base Design Techniques: Requirements and Logical Structures*, S. B. Yao, S. B. Navathe, J. L. Weldon, and T. L. Kunii, Eds. Springer-Verlag, Berlin.

YORDON, E. 1982. *Managing the System Life Cycle*. Yourdon Press, New York.

Received June 1986; final revision accepted May 1987.