Information Systems Failures – a Survey and Classification of the Empirical Literature

2 authors:

Kalle Lyytinen
Case Western Reserve University

Rudy Hirschheim
Louisiana State University

Oxford Surveys in Information Technology, Vol. 4, 257–309

Information systems failures—a survey and classification of the empirical literature

KALLE LYYTINEN Department of Computer Science, University of Jyvaskyla, Jyvaskyla, Finland

and RUDY HIRSCHHEIM Oxford Institute of Information Management, Templeton College, Oxford

Contents

Abstract

A number of studies report that at least half of all information systems are failures—a situation which poses a considerable challenge to the information systems community. This paper discusses how this challenge has been addressed in the literature. In particular, it summarizes how the notion, types, and causes of information system (IS) failure have been discussed in the literature. The paper shows that the widely employed concepts of IS failure are vague. Three principal failure notions are identified in the literature, but all suffer from conceptual weaknesses. To overcome these problems the paper proposes a new notion: that of an expectation failure. It defines the concept of IS failure pluralistically as a problematic situation confronted by a particular IS stakeholder group. The paper classifies major problems that are met when developing and using information systems, and demonstrates that most of them are social and political in nature. Twelve classes of reasons for IS failure are drawn from the literature. At the end several weaknesses in the empirical literature dealing with IS failures in the past are identified. The paper concludes with some recommendations for both IS practice and research.

1 Introduction

Developing computer-based information systems is never easy, nor are they always used effectively. Schmitt and Kozar (1978) provide a good example of a major system which clearly met its design objectives, but could hardly be considered successful. The system, although technically very elegant, was hardly used by its end-users—they were never convinced of its value—yet the system continues to remain in existence. This is a classic phenomenon. Systems live on even though they are not used. Keider (1978) captures the dilemma succinctly when he writes: '. . . projects never terminate. Rather, they become like Moses, condemned to wander till the end of their days without seeing the promised land.' Writers such as Mowshowitz (1976) have gone so far as to state that many, if not most, information systems are failures in one sense or another. Failures in information systems development (ISD) and use are legendary, and have been widely discussed over the past two decades. Classical examples can be found in Brooks's (1974) discussion on 'the mythical man-month', and Lucas's (1975) empirical analysis of 'why information systems fail'. IS failures continue in the 1980s. (Until recently, there was even an entire journal devoted to the discussion of failure: *Systems, Objectives, Solutions*.) According to a recent survey (Gladden 1982), 75% of all systems development undertaken is either never completed or not used if completed. Several writers (Canning 1977; Lientz and Swanson 1980) point out that an inordinate amount of systems development funds (70%) is spent on systems 'maintenance'. Taken in total, one is led to seriously question the state of IS development practice. Some have gone so far as to refer to it in terms of a 'crisis in systems development' (cf. Sibley 1986; Bubenko 1986; Martin 1985).

Clearly, it is important for the IS community to better understand IS failure. But whilst there exists a wealth of literature on this topic, the notion itself has remained somewhat nebulous and ill-defined. In fact, most studies of IS failure are based on a largely unarticulated failure concept. Even those attempts to explore the concept in a more systematic way (Morgan and Soden 1973; Lucas 1975; Sanders 1984; Bailey and Pearson 1983; Simila and Nuutinen 1983; Robey and Markus 1984; Ginzberg 1980) attach a too limiting concept to failure. They neglect the multiplicity of reasons for, and multi-causal nature of, IS failure. They also miss the rich varieties of IS failure.

Because of the inadequate conceptual clarity of the IS failure notion, it is not surprising that there is currently no sensible framework to understand and analyse IS failure. The purpose of this paper is to offer a new IS failure concept, and an associated framework for better understanding IS failure. The framework categorizes the main types of IS failure and explores the reasons for failure. It helps to identify the richness and variety of IS failure.

The paper is organized as follows. In Section 2, we introduce a new IS failure concept: an expectation failure. In Section 3, we discuss three widely accepted concepts of IS failure, and critically evaluate them with the help of the expectation failure notion. Our evaluation shows that whilst there are many notions of failure, they are somewhat limited and suffer from several conceptual and methodological weaknesses which make them unsuitable for a more systematic study of the phenomenon. In Section 4, sixteen classes of IS failure are identified and discussed. The reasons which lie behind failure are explored in Section 5. Section 6 provides some thoughts on the implications for research and practice.

2 Stakeholder model of IS failure

The first impression one gets from even a quick a perusal of the literature on IS failure is the lack of any detailed treatment of the failure notion: what precisely is meant by 'failure' and what is meant by 'success'. This is hardly a trivial matter, yet few have bothered to explore it in any detail. Many in the IS community (both researchers and practitioners) seem to consider IS failure as self-evident, requiring no further clarification. Either an IS succeeds or it fails. When it is discussed in the literature, it is not identified as a singular constellation of facts, but as a family of situations in which ISs have been found to fail. Some typical examples of IS failures are: when the potential benefits of the IS are not realized (Alter and Ginzberg 1978), when the IS is not used (Lucas 1975), when the users' attitudes are negative (Bailey and Pearson 1983), when there is substantial user resistance (Markus 1983), or when a functioning system is not delivered (Gladden 1982). Each situation identifies only one aspect of failing. This raises a series of questions which any study of IS failures must address: what are those situations in which an information system can be said to fail? are they the only conceivable ways in which an IS can fail? are they really failures or just symptoms of more substantial difficulties? how do different failures relate? These questions are explored in some detail in Section 3. Here, we concentrate on the first question: how can we analyse those situations in which ISs fail?

2.1 IS evaluation and IS failures

IS failure is an endemic part of the larger issue of information systems evaluation. Any study of failure must concern itself with the derivation and analysis of the value of an information system. Accordingly, whatever can be said about evaluating IS success can also be used for understanding IS failure. Yet, there appears no single definition of IS success, nor is the process of accurately measuring success clear. A number of writers (Legge 1984; Blackler and Brown 1986; Hirschheim and Smithson 1986) go so far as to question the very basis of what 'success' implies. Further, any IS evaluation intervenes into an organizational situation, which means that failure assessment is an organizational process that affects the participants and their surroundings, i.e. it is 'political'.

There are, however, some differences between IS evaluation (in general) and IS failure assessment (in particular) which suggest IS failure studies have their own emphasis and approach. First, the reasons for failure and success do not necessarily coincide: there is not only a quantitative, but also a qualitative difference. For example, the acquisition of a more powerful computer may remove a performance bottleneck, but this does not necessarily make the IS successful. Success might require changes in user education, organizational arrangements, or even the abandonment of the IS altogether. Second, traditional IS success assessment emphasizes more 'summative' evaluation of the IS (i.e. what occurred), whereas failure studies focus more on the reasons why something occurred, i.e. 'formative' evaluation (cf. Weiss 1972).

2.2 Values and IS failure

As suggested above, IS failure is a complex phenomenon. The term 'failure' has different senses depending on the specific perspective one adopts: 'failure' conveys a family of meanings. Accordingly, the notion of IS failure must involve a pluralistic component that takes into account the rich variety of existing perspectives (Kling 1980; Hirschheim 1986). The roots of such a notion must therefore lie in social values (Ulrich 1983). The values are introduced in IS by goals and standards which state in more concrete terms what should be achieved and how. Values are, however, often conflicting and ambiguous. This creates the need for pluralism in understanding failures.

It goes without saying that failure assessment only makes sense if values, and how they enter into an IS, are clearly understood. In general, we shall assume that values are social: they provide evaluative dispositions and guide social behaviour by representing inter-subjective preferences and ideas. What values we have depend more on our social and cultural inheritance than on our personality traits and individual utility scales. Were it not for socially shared values, IS failure assessment would be a completely subjective process with no common basis for discussing failures. Each failure would be its own Odyssey. Social values also determine the criteria on what to evaluate: they pinpoint those aspects that are worth the cost and effort of assessing. Without a common pool of social values there would be no point in developing frameworks and tools for assessing and analysing IS failure[1].

Using the language of values, we posit that IS failures occur when outcomes that would coincide with adopted values are not obtained, i.e. some goal is not achieved or performance is below standard. But as values can be conflicting and ambiguous there can also be different points of view about failure.

2.3 Stakeholders and IS failure

Because values are largely constituted in the realm of a community rather than in the realm of the individual, finding values that are relevant for IS design implies a need to identify a group of people (in the extreme case only one person) sharing a pool of values that define what the desirable features of an IS are and how they should be obtained. We shall call such groups of people *stakeholders*. According to Mason and Mitroff (1981), stakeholders are defined as 'all those claimants inside and outside the organization who have a vested interest in the problem and its solution' (p. 43). We thus assume that there are no 'general' IS failures; rather, IS failures pose problems for someone or some group. They signify a gap between some existing situation and a desired situation for members of a particular stakeholder group.

But this does not describe how stakeholders' values come about. It is our contention that stakeholders' values originate mainly from the interests various stakeholders try to pursue. They represent evaluative orientations that advance stakeholders' interests. In general, an interest is something which furthers a stakeholders' advantage. It derives mainly from a group's position in the organization and from associated rewards and power bases. Consequently, stakeholder groups attempt to control organizational resources which enable them to channel organizational action to outcomes that advance their interest. Interests are vested, although in many cases several stakeholders (or all) can share an interest. Stakeholder groups, thus, can possess a common interest[2].

But how are stakeholder groups and their values identified? Clearly, the identification of an interest (or stake) is not much easier than the perception of their shared values. Difficult questions arise:

[1] On a more abstract level, social values can be said to convey design ideals, i.e. general and internally consistent evaluative orientations (Kling 1978; Klein 1985) although in every-day life, pursued values can be fragmentary, inconsistent, and remain largely unconscious.

[2] The stakeholder concept is necessarily a dynamic one, and cannot easily be distinguished from the notions of agency (i.e. potential for action), and power (i.e. capability to employ various resources to secure stakeholders' sectional interests) (Giddens 1984). Stakeholder groups often organize themselves to increase their influence. Values, which are frequently ideological, surface for such organized groups to rationalize and justify their action.

is everyone who thinks or believes that he or she has a stake in some situation a stakeholder? And what about those people who do not see themselves as stakeholders although some others may think so? This is a point where conceptual confusion exists in the literature. Some studies see stakeholder roles as depending on the actor's belief that he or she has a stake (Mason and Mitroff 1981). In this case the criterion of stakeholder identification is the content of the actor's beliefs. Others, notably Marxists and Weberians, focus more on the structural features of the situation, identifying stakeholder groups independently of what the actual actors think[3]. In this case the identification criteria are observed structural arrangements and associated interests in the social space of actors.

In this paper, we adopt a structural approach and relate the stakeholder concept to the structural properties of IS design and use. We adopt such a position because we believe that ISs are ostensibly institutional arrangements that possess structural similarities to other social institutions (Kling and Scacchi 1982). Therefore, the stakeholder groups are mainly derived from the more enduring social structures in ISs which arrange relationships between actors in a social space.

Irrespective of the standpoint one chooses with which to identify the stakeholders, the task is still far from simple and it cannot be accomplished with any clear-cut classification scheme. In the literature, IS stakeholders fall into three main groups: users, management, and IS professionals. Unfortunately, this classification is much too coarse and, in most cases, inadequate, as it conveys the role prescriptions associated with the design of an IS. It does not reveal the actors' actual interests with regard to IS; instead, it focuses on intended and observable aspects, ignores conflicts inside these three groups (cf. Markus 1983; Kling and Iacono 1984; Franz and Robey 1984), and provides a much too simplistic view of the IS and how it affects an organization's members' interests.

In an attempt to identify IS stakeholders, a list of four criteria is offered which, it is hoped, provides more guidance in distinguishing key stakeholder groups. The criteria are:

(1) nature of IS,
(2) type of relationship to IS,
(3) depth of impact,
(4) level of aggregation.

The first criterion ('nature of IS') underlines the fact that IS should be seen from many perspectives. The IS might be seen as: a sophisticated piece of technology, a communication tool, a decision-making tool, a control mechanism, and so on. Within each perspective we may distinguish different groupings of IS stake-

[3] Of course, the actors in many cases are cognizant of their interests, but this is not necessarily so; (in Marxism this is called 'false consciousness').

holders. The 'types of relationships' might include: producer, consumer, sponsor, regulator, user, etc. The 'depth of impact' may be direct or indirect. The 'level of aggregation' may vary from one situation to another: from distinguishing between individuals (one actor as a stakeholder), and groups (multiple actors as a stakeholder), to larger collectivities such as a company or a society.

In an inventory control system, for example, the list of four criteria might result in the following stakeholder map. The 'nature of the IS' could result in a number of relevant perspectives. Under a technical perspective, we might identify technical staff, designers, vendors and so on; from a decision-making perspective, different decision-making units could be highlighted as well as how the system might affect decision-making power (cf. Kling 1980; Bjorn-Andersen and Pedersen 1980); from a communications perspective, we might note communicating social groups and actors, as well as how the communication patterns are changed due to the introduction of the system. In terms of the 'types of relationships', it is possible to identify systems analysts, consultants, and end-user representatives as producers, specific individuals as consumers, senior management as the sponsor, and so on. The 'depth of impact' could be confined to direct effects, where the system directly affects particular users. The 'level of aggregation' could specify either individuals (for example career prospects, salary), groups (working patterns, socialization) or whole formal aggregates (the company).

Clearly, the number of stakeholders is likely to be considerable in real-life IS.

2.4 Expectation failure

Stakeholders' interests are formulated through a number of expectations, i.e. the beliefs and desires concerning how the IS will serve the group's interests. Expectations represent evaluative dispositions which are derived from the stakeholders' common pool of values. In many cases these expectations are vaguely expressed, and are never rationalized or verbalized as real concerns because of: the great number of stakeholders; stakeholders' inability to voice their expectations because of organizational barriers, dominating ideology, lack of time or interest; or simply the unclear content of the expectation. Some of the expectations, however, are always formulated explicitly to justify the effort and investment in the IS. These are described in terms of systems requirements, goals, and design standards.

We are now in the position to define information system failure as the 'inability of an IS to meet a specific stakeholder group's expectations'. Failure does not therefore involve only the system's inability to meet requirements, as only a fraction of stakeholders' concerns are usually formulated in them. (Many IS failures actually have met the requirements, but they have been considered failures because some other vital concerns have not been catered for.)

It is important to note what our definition says and what it does not say. First, the definition does not say how the failure is detected nor how it is resolved. This will vary from one failure to another depending on how the stakeholder group perceives the situation and how it organizes action to remove it (if at all). Second, the definition does not say how many groups are affected by the failure. Depending on the type of failure, it may concern one or many stakeholder groups—but usually in different ways. Third, the definition does not say how failures come into existence. There are usually two ways in which the failure comes about: *structural* and *processual*. In the former, an IS design affects the structural properties in a stakeholder's environment such that it clashes with the stakeholder's interests. In the latter, a stakeholder's level of expectation may change as a result of organizational learning, persuasion, political campaigning, and the like. In this case, the stakeholders learn to see the IS as fitting incompatibly with their interests. Both are important and affect how the stakeholder's action is organized around the IS. In this sense IS failure is at the same time a starting point and a result of stakeholders' action by which they try to realize their interest. Fourth, the definition does not say how the IS is to be perceived and what means are appropriate for perceiving it. Instead, the concept of an *expectation failure* stresses the importance of understanding how various stakeholders perceive and comment on the value of the IS: failure is the embodiment of a perceived situation. Thus, the notion of an expectation failure provides considerable freedom in terms of how the failure is detected, how it affects various stakeholder groups, how it comes about, and how it is perceived.

3 IS failure in the literature

It is our contention that the literature on IS failure can be classified according to how the notion of failure is defined. There are four major categories:

(1) correspondence failure,
(2) process failure,
(3) interaction failure,
(4) expectation failure.

As expectation failure is discussed in some detail in the previous section, we concentrate in this section on the other three. These failure notions have been widely adopted in the past and could be considered as the traditional failure concepts. Our exposition of these concepts is done by interpreting the three other failure notions as special instances of an expectation failure, reflecting specific interests of powerful stakeholder groups.

3.1 Three traditional failure concepts

3.1.1 Correspondence failure

Correspondence failure is the most popular notion of IS failure, and typically expresses management's view of IS failure (Cooper and Swanson 1979; Alter and Ginzberg 1978). Its main premise is that design objectives are stated in advance, and if these are not met, the IS is a failure—hence the name 'correspondence failure'. It is generally believed that the design objectives are objective and formal, and that their achievement can be accurately measured. For this reason failure assessment is thought to employ managerial control methods such as cost–benefit analysis and cost-accounting techniques.

In practice, correspondence failure is often found to be too idealistic to be of much value—even as a managerial strategy. It provides the ultimate rational ideal of how failure should be viewed. Consider the following. First, there have been ISs which met their objectives, but when viewed globally could hardly be construed as successful (Schmitt and Kozar 1978). Second, many researchers (cf. March and Olsen 1976) have pointed out that management objectives are ambiguous, only expressing broad hypotheses for action. For this reason objectives tend to be fragmentary, conflicting, and their interpretation is context-driven. Objectives can also be ambiguous due to the low quality of, and bias in, requirements specification (Cooper and Swanson 1979), or due to conflicts in management perception and associated bargaining (Keen 1981; Kling and Iacono 1984). In many cases it is difficult, if not impossible, to observe objectively any correspondence. Third, even if it were possible to accurately represent management objectives, the difficulties in accurate measurement of IS performance are far from trivial (Mason and Swanson 1981). In fact, the problems in verifying and validating measurement instruments are immense (Hirschheim and Smithson 1986).

3.1.2 Process failure

In many situations when the IS cannot be produced within given budget constraints, it results in what is called a 'process failure' (Turner 1982; Brooks 1974; Gladden 1982). The concept of process failure captures two related but distinct aspects of unsatisfactory performance in producing the IS. First, if the IS development process cannot produce any workable system, it is without doubt a failure. Usually this would involve unresolvable problems in designing, implementing, or configuring the IS. Second, and the more common aspect of process failure, is when the process produces an IS, but one which involves vast amounts of overspending both in cost and time, thus limiting or negating the global benefits of the system.

Process failure signals management's inability to achieve predicted resource allocations, to develop appropriate costing and budgeting schemes for systems design, or to predict the implementation difficulties of information technology. The literature on software economics and software engineering is largely informed by this view. One outcome of the overspending is the sheer size of the systems backlog for applications development, which in many organizations is in excess of two years (Martin 1985).

3.1.3 Interaction failure

Because of the problems in measuring IS objectives, some researchers (Lucas 1974, 1975; Robey 1979; King and Rodriguez 1978; Robey and Zeller 1978) have suggested that a low level of IS use can be used as a surrogate for IS failure. Some related measures are user attitudes (Swanson 1974; Schewe 1976; Maish 1979; King and Rodriguez 1981) and user satisfaction (Pearson 1977; Bailey and Pearson 1983). The argument is simply that if end-users properly and intensively use an IS, it implies that they are satisfied with it, that their attitudes are positive, and that their task performance is improved (Ginzberg 1980). If, on the other hand, the interaction is low then the attitudes are negative with decreased task performance. Because primary attention is given to users' interactions with the system, this failure concept is termed 'interaction failure'.

Unfortunately, there is little empirical evidence supporting the claim that heavy IS use correlates with high user satisfaction and improved task performance. And the existing evidence is largely controversial (cf. Zmud 1979; Srinivasan 1985). For example, Lucas (1975), Lucas and Plimpton (1972), and Keen (1975) found a positive correlation, but Schewe (1976) found no significant correlation. Robey (1979) noted a strong correlation, but Srinivasan (1985) found that the relationship was not always positive. In addition, the concept of interaction suffers from several other difficulties. The empirical literature abounds with examples of systems which were heavily used but were not perceived to be successful in the eyes of some users (Kling and Iacono 1984; Markus 1983). Very often the reason is that there is no real alternative (Ginzberg 1975; Keen 1975).

There are also conceptual problems because the notion of IS use is far from clear. Normally, IS use is measured by the number of interactions with, or the volume of data transferred from, the system. But these measures are unsatisfactory. Does the time used interacting with the systems or the amount of data transferred between the system and the user really show how the system is used or how it contributes to task performance? The answer is probably no, as the time used can also depend on the type of interface or some other factors, and the amount of data transferred heavily depends on prior design decisions. It is also possible to conceive of successful systems which are used only once (when they are really needed). The problem with the above measures is that they do not express what the users *do* with the system. Several studies have shown that the use of the system can be seen from many angles: it has symbolic, political, and persuasive dimensions in addition to a decision-making dimension (Feldham and March 1981; Boland and Pondy 1985). Thus high interaction is not necessarily indicative of improved task performance, rather it can indicate to what extent various IS stakeholders employ the IS to sustain or increase their bargaining position inside the organization.

3.2 IS failure notions—relationships and evaluation

Table 1 summarizes the key concepts of the four failure notions.

3.2.1 Relationship of the failure notions

The first three notions in Table 1 are strongly related to each other despite their surface differences. This is illustrated in Fig. 1. For example, process failure correlates with correspondence failure as high development costs undermine global organizational benefits. The high cost and long system development time may also negatively shape users' attitudes to the system and decrease its use.

Table 1. Four failure notions

Failure type	*Content*	*Problems*	*Literature*
Correspondence failure	IS does not match goals; criteria: IS quality criteria	• IS has failed although it achieved goals • ambiguous, conflicting goals • measurement • static	Alter and Ginzberg (1978) Ein-Dor & Segev (1978)
Process failure	IS not designed in time and cost; criteria: budget limits, cost and resource allocation schemes	• no measure for utility • emphasis on efficient resource allocation	Gladden (1982) Turner (1982)
Interaction failure	IS is not used; negative attitudes; criteria: user attitude surveys, interaction statistics	• mandatory use • use measures too unsophisticated • non-use may be reasonable • measurement can bias results	Lucas (1978) Pearson and Bailey (1983)
Expectation failure	IS does not fulfil expectations; criteria: values of IS stakeholders and their perceptions	• bias • halo-effects • accuracy in assessing expectations • manipulation of prejudices	Checkland (1981) Robey (1984) Kling and Iacono (1984)

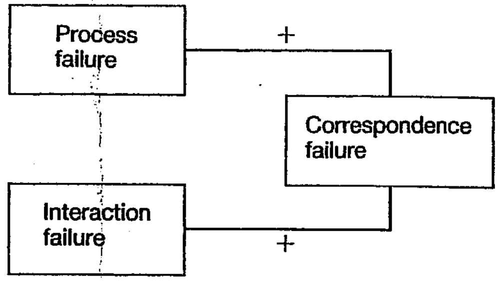

Fig. 1. Relationships among the three traditional failure notions.

The three notions together can be seen to express interests and values of those stakeholders who principally gain from the IS and who are directly involved in specifying and managing computing arrangements as principal producers, sponsors, and so on. The three notions portray a highly rational image of IS design, and view IS mainly as a neutral technical artifact and tool.

On the other hand, the three failure notions are not necessarily correlated if a more pluralistic view of IS is adopted. If a process failure occurs in a situation where the information system is not in accord with management policy, it does not necessarily contribute to a correspondence failure. Moreover, employees may in some situations heavily use the IS to reduce managerial control (Conrath and duRoure 1978). Hence, high interaction does not always mean that the IS is used to achieve managerial objectives.

The expectation failure notion is a superset of the other three. It conveys a more encompassing, politically and pluralistically informed view, which helps to analyse relationships among the other failure notions and to understand their social origin. In this way the other failure notions can be 'reconstructed' by using the expectation failure concept.

3.2.2 Comparison of failure notions

The concept of IS failure is important in that it guides the perception and evaluation of an IS; it affects how the evaluation process is conducted. The four failure notions thus provide the scope, goals, methods, and process of IS failure evaluation. In what follows we compare the alternative failure notions using eight criteria:

(1) dimensionality,
(2) type of measurement scale,
(3) nature of assessment,
(4) temporal aspect of assessment,
(5) nature of assessment technique,
(6) assessment time frame,
(7) assessment participants,
(8) nature of assessment process.

(*1*) *Dimensionality*. This criterion denotes the number of aspects in the IS to be considered during failure assessment. The number of dimensions can range from one to several. The first three failure notions apply only one or two dimensions in observing that an IS has failed, and, hence, see failure in a quite simple way. For example, process failures concentrate only on the time and cost measures of software development and the technical obstacles causing the overspending. By so doing, they disregard all other aspects of 'failing', such as resistance, non-use of the software, the non-achievement of management objectives, and so on. In a similar vein, interaction failures ignore all problematic aspects related to IS development.

Expectation failures, on the other hand, are multi-dimensional. Any situation which fails to meet expectations of any stakeholder group can be regarded as an IS failure. These can involve technical, economic, psychological, behavioural, and political aspects. Moreover, the IS can fail to realize these expectations in many ways. Sometimes IS failure can take place in a very short time as, for example, when implementers face sudden user resistance. Alternatively, the failure can be a slow, hidden process that is pervasive over a longer period of time as, for instance, a slow decrease in performance and response times. Additionally, the scope of failure can vary. Some failures have dramatic and broad consequences, like catastrophes in nature. Programming errors which result in totally erroneous behavior are examples. On other occasions, failure can be small in scope and depth. It can be concluded that failures need to be studied in a more systematic manner, requiring us to be more careful in delineating different types of IS failures.

Dimensionality also involves how the perceptions about a specific aspect are employed. Here we can distinguish between *summative* and *formative* dimensionality. Summative assessments produce only evaluations: they tell how good or bad some aspect of the IS is. Formative assessments provide diagnostic information which helps to explain why a specific situation was reached. All failure concepts can use both summative or formative assessments. In the first three, however, the main emphasis tends to be on summative evaluations; conversely, in the expectation failure it is the formative one which is preferred.

(*2*) *Type of measurement scale*. This criterion is the measurement scale used in failure assessment, and can be either *discrete* and *continuous*. On a discrete scale, when a value of a specific factor is increased to a specific point, success turns into a failure and vice versa. When the scale is continuous, there is a gradual shift from 'poor' to 'good' or failure to success.

The three traditional failure concepts are founded on a discrete measurement scale. For example, if the interaction with the IS is higher than a pre-specified standard then the IS is considered a success. In this sense, these failure concepts suggest that most IS failures resemble catastrophes which result from adding the proverbial 'last drop'.

The expectation failure notion assumes, instead, that IS failure and success form a continuum where the level of fulfilling a stakeholder's expectations ranges from very low to very high. Because IS designs often represent political compromises or involve incomplete designs, ISs are typically neither prominent successes nor major failures. As Kling and Iacono (1984) point out (p. 1225):

> 'Which is "successful" from the standpoint of implementation concepts can be problematic for its users. We often find that practitioners are embarassed about systems which have imperfect architectures. We suspect that such systems are common, but need not be viewed as "failures".'

(3) Nature of failure assessment. This criterion distinguishes between *neutral* and *non-neutral* assessment. If IS failure is regarded as neutral, it is an unintentional accident and hurts all stakeholder groups' interests equally: it has no re-distributive effects. Alternatively, if the failure is non-neutral, then the IS can have re-distributive effects—some of them intentionally produced.

The three traditional failure notions perceive IS failure as largely neutral, whereas the expectation failure notion sees IS failure assessment to be value-laden and non-neutral (i.e. some stakeholder groups lose and others win through information systems development). In this case, the policy implications of a 'failure' are unclear and conflict laden. Markus's (1983) study of the use of a financial information system provides a good example. She found that some end-user groups resisted the system whereas others supported it and its further development. The outcome was continuous political campaigning and strategic action by both user-groups to make their cause triumphant. These differences in perceptions of the value of IS were linked to the clashing structural interests each group had in the organization and how the designed IS enabled them to be pursued.

(4) Temporal aspect. This criterion embodies the temporal dimension in failure assessment. There are two alternatives: *static* or *dynamic*. If a static view is chosen, IS failure is regarded as a state—a thing to be observed. By implication, IS failure will have some temporal endurance; once an IS has failed it will remain so. When a dynamic view is chosen, IS failure is seen as a dynamic process that is reshaped by stakeholders' activities, perceptions, and learning.

The three traditional failure notions prefer the static view. For example, if objectives have not been met by the IS, or user attitudes are negative, this indicates that the IS is a failure.

In the expectation failure notion, the temporal dimension is dynamic. A failure is a stakeholders' perception that will be actively shaped by their future activities. IS failures are dynamically acted upon by actors involved in the situation. Failures are reacted to by political moves, changes in the organizational climate, persuasion, technology changes and so on. A strong case for a dynamic view is Keen's (1981) discussion of implementation and counter-implementation strategies.

(5) *Nature of assessment procedure*. This criterion singles out the properties of the assessment procedure. If a well-defined technique is applied in a step-by-step manner, failure assessment is *formal*. The assessment procedure is *informal* if no strict rules are applied and the process to a large extent relies on intuition, personal skills (tacit knowledge), and more unstructured expressions of opinions. The informal and formal are the two extremes of a continuum.

The three traditional failure notions embrace more formal assessment procedures. For example, the techniques of cost–benefit analysis and risk analysis form the backbone of the correspondence failure. Similarly, the process failure concentrates on project management techniques, software quality assurance, and the like (Boehm 1981).

The expectation failure notion views failure assessment from the context of organizational bargaining and hence adopts more informal assessment procedures. Since failure is a perception shaped by the political situation and actors' goals, it always involves a dynamic element, and no closed collection of rules can exhaustively deal with it. Therefore, the assessment procedure acts more like an agenda or protocol for inquiry and negotiation. More formal techniques enter this protocol in explicitly defined stages to help in substantiating claims made, or in helping to analyse some observations of the situation. They, however, never replace the actual bargaining, interpretation, and inquiry processes by which the situation is observed, analysed, and resolved.

(6) *Assessment time frame*. This criterion expresses the time horizon of the assessment, and asks whether assessment looks primarily to the past, present, or to the future. Accordingly, assessments fall into two main classes: *ex ante* and *ex post* analyses.

The four failure notions can all be used to accomplish both *ex ante* and *ex post* analyses. However, their primary time horizon varies. The traditional failure notions tend to adopt an *ex post* focus. For example, correspondence failures look back on how the objectives have been achieved, process failures emphasize budget control and monitoring of time plans, and interaction failures focus on the existing level of interactions, users' satisfaction, etc. In expectation failures, the focus is on the future, i.e. in reshaping the organizational context and technical implementation of the IS to fulfil the expectations of the stakeholder group.

(7) *Participants*. Failure assessment takes place in a specific organizational context, where tasks and responsibilities to carry out the assessment process are assigned to several organizational roles. Different failure notions, due to their intrinsic differences, prescribe different sets of organizational roles to failure evaluation. Here we can distinguish between *limited* and *extensive* role sets. In the

former, the evaluation effort is assigned only to a few, and in some cases only to one organizational role. In the latter, the set of roles is open-ended and determined dynamically by the clashing interests of stakeholders. The traditional failure notions advocate a limited set of roles. For example, correspondence failures prescribe tasks and responsibilities to management and analysts. The expectation failure notion, alternatively, assumes an extensive role set. All the IS stakeholder groups may assume many roles in the evaluation effort.

(8) *Evaluation process.* The last criterion addresses the context or features of the evaluation process. Here we can distinguish between *context-independent* and *context-dependent* evaluations. If the failure assessment is context independent then the process outcome is claimed to be independent of the evaluation context. That is, given data about the problem situation and the failure resolution approach, the outcome of failure assessment will be the same in all contexts and is unaffected by the participants' dynamic behaviour. If a context-dependent evaluation process is presupposed, then all results of failure assessment should be understood in relation to the features of the evaluation context, i.e. stakeholders' dynamic behaviour, their interpretations of the situation, and so on. The evaluation outcomes do not exhibit the same type of scientific objectivity as in the context-independent alternative. The three traditional notions tend to adopt a context-independent view of the evaluation process, whereas the expectation failure notion advocates a pluralistic and contextual view of the evaluation process.

Table 2 summarizes the results of the comparison.

Table 2. Comparison of IS failure concepts

IS failure criterion	*Correspondence failure*	*Process failure*	*Interaction failure*	*Expectation failure*
Dimensionality	mainly one; summative/formative	mainly one; summative/formative	mainly one; summative/formative	several; formative/summative
Measurement scale	discrete	discrete	discrete	continuous
Nature of failure	neutral	neutral	neutral	non-neutral
Temporal aspect	static	static	static	dynamic
Assesment procedure	formal	formal	formal	unformal
Evaluation time frame	*ex post/ex ante*	*ex post/ex ante*	*ex post/ex ante*	*ex ante/ex post*
Participants	limited	limited	limited	extensive
Evaluation process	context independent	context independent	context independent	context dependent

3.3 Summary and discussion

Our analysis of the four failure notions suggests that the correspondence, process, and interaction failures, apart from their surface differences share a common background: a highly rational view of IS failure (cf. Hirschheim and Smithson 1986). For this reason, the three concepts tend to ignore the underlying value orientations, the larger social context of any evaluation, and the conflicting and ambiguous uses of failure assessments (in particular their unclear policy implications). Correspondingly, the research guided by these three notions has focused on methods and tools for failure assessment, i.e. on means of problem identification and resolution. Their view of failures is, accordingly, based on causal theories of social and organizational behaviour.

The expectation failure notion, on the other hand, focuses on the underlying values and on the cultural, economic, and historical factors that make these values relevant for some IS stakeholder group. Since these values differ, the expectation failure notion is necessarily pluralistic. It is also political, as it sees many of the values to be in conflict and their resolution to occur through political action. Failure assessment is seen as an integral part of a larger social context. Expectation failure places a high premium on the understanding of organizational arrangements in which failure assessments take place, and on the evolutionary nature of evaluation processes. It adopts a complementary view of IS failure and provides a different 'story' to view IS design and use (Robey 1984).

Our analysis makes no normative statement about which failure notion is 'best'. Which concepts (or a set of concepts) are 'best' depends on the values and preferred framework of the inquirer. But in choosing which failure notion is most appropriate for our analysis, we adopt what Klein and Hirschheim (1983, 1985) term a 'consequentalist perspective' (see also Ulrich 1983). This perspective suggests that the choice should be based on the practical consequences of adopting a particular concept. The following criteria could be used.

(1) How does the concept help in perceiving various IS problems?
(2) How does the concept help to understand factors and reasons that bring about the problematic situation?
(3) How does the concept help to identify groups of people concerned?

In all respects, the expectation failure notion appears better. First, it is multi-dimensional and can thereby accommodate many kinds of problematic situations. It urges us to study IS failure in multi-dimensional terms—a subject covered in more detail in Section 4. Second, it stresses formative, *ex ante*, and contextual aspects of failure perception and resolution—all necessary for better understanding the root causes of failure. Third, by definition the expectation failure notion covers all IS stakeholders' concerns—many which are not identified in other failure notions. Its view of an IS stakeholder is also dynamic; in the other failure notions, the stakeholders are pre-specified.

A negative consequence of adopting an expectation failure notion is that, being a pluralistic concept, it involves a high level of complexity and does not lend itself to any straightforward procedure. It also denies the availability of simple failure resolutions in all situations—every policy can be a self-defeating one. However, these caveats only highlight that IS failure is an extremely complex web of social and technical phenomena. All notions which fail to appreciate this complexity, or try to reduce it to technical complexity, are likely to be too crude to be of much value.

4 Classes of IS failure

Although the notion of an expectation failure helps us to understand the concept of 'failure', the set of IS failures is fundamentally open-ended. This necessitates a more thorough exploration of the ways in which an IS can fail and how these failures relate to various stakeholders' concerns. The principal goal of such an exploration is to understand how IS affects stakeholders' interests and to classify the impacts of IS design and use. In this section we shall develop a preliminary framework in which to classify various IS failures as well as to explore their relationships.

4.1 IS failure classification framework

Although there is a wealth of literature on IS failures, few studies have attempted to categorize them or to suggest frameworks for such classifications. A notable exception is Alter (1980), who divides IS failures into several distinct classes, ranging from technical problems to conceptual problems. The weakness of Alter's classification is twofold. First, it is not based on any systematic conceptualization of the field, but rather on an *ad hoc* list of IS problems drawn from his consulting experience. Second, his classification is not exhaustive and ignores a number of important problem areas. Other proposed classification schemes (e.g. Kling 1980; Markus and Robey 1983; Lucas 1975, 1981) suffer from similar weaknesses.

In an attempt to overcome these weaknesses we propose an IS failure framework based on two dimensions. The first dimension discloses some *aspect* of an IS that becomes problematic; the second identifies the *temporal* feature of the problematic aspect in the IS life-cycle[4].

The framework proposed is quite general, yet powerful, and is based on the broad notion of social action (Etzioni 1967; Habermas 1984). In the first dimension, the aspects distinguished are the four domains of IS: the *technical* domain, the *data* domain, the *user* domain, and the *organizational* domain. The domains are interrelated and form a layered structure depicted in Fig. 2. Every domain consists of a set of distinct phenomena and principles governing

[4] We shall attempt to keep the number of IS aspects small. At the same time, we want the aspects to be disjoint and exhaustive. Our goal is to have a framework which allows a pluralistic evaluation of each aspect. General IS evaluation frameworks (Waters 1974; Iivari 1983; Hamilton 1981) are unacceptable as they are neither pluralistic nor contain a disjoint classification scheme.

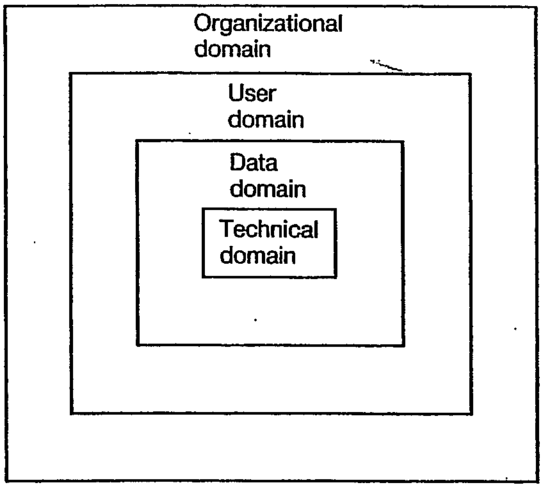

Fig. 2. Four domains of IS failure assessment.

their behaviour and evolution. The technical domain includes the physical means and technical know-how by which computing tasks are carried out, i.e. information technology, storage media, and tools and methods to design and implement computer systems (Bakopoulos 1985). The data domain deals with the nature, form, and content of the data processed by, and communicated in, the system. The user domain identifies the skills, competence, personality traits, and motivational factors of the user population that directly or indirectly are involved with the system. The organizational domain captures the nature and content of organizational roles by which organizational activities and tasks are carried out. The focus is on behaviour expectations, responsibilities, authority, and task performance of an organization's members.

In an information system, these domains interact and are under constant change. The organizational domain, for example, largely determines the size and level of the user population, and the nature and content of the data. It is also an important parameter in technology choice (size, power, and nature of technology used). It is important to note that the domains interact as if they were layered, with interactions taking place in all directions of the layered structure. For example, it is possible to look at how technology affects organizational designs, authority structures and decision-making processes, and so on.

In the second dimension, we consider the temporal feature of IS; that is, how the four domains of an IS charge over time. When the interactions between the four domains are relatively fixed and are not the focus of organizational action, we speak of *IS use*. When the domains and their interactions become the focus of organizational action and are seen as the targets of change, we speak of *IS development*. In IS development the phenomena in the domains and

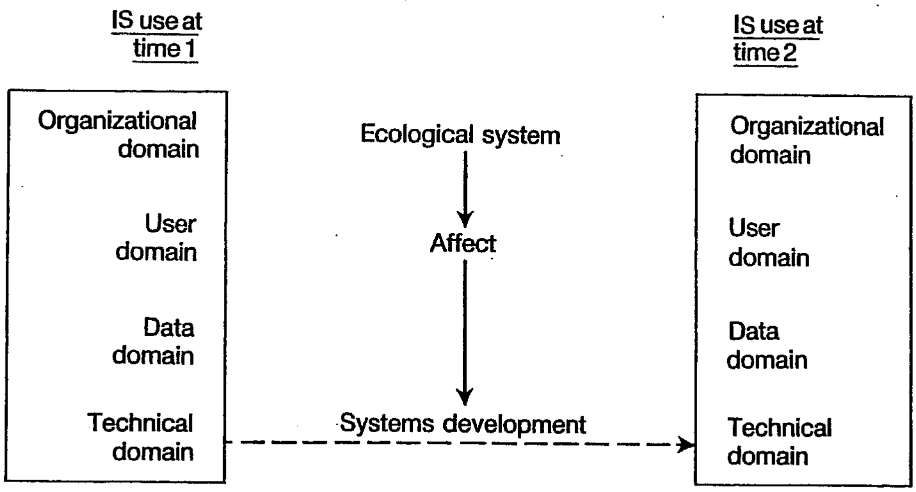

Fig. 3. IS development and use.

their interactions become problematic and are seen to be in need of change (Lyytinen 1987b). Figure 3 attempts to portray the distinction between IS development and use.

The framework for classifying IS failures can now be developed using the two dimensions as follows. We first observe those elements of an IS that make up the four domains. Then we analyse how these elements interact or are pertinent to some stakeholder's interest. In other words, we observe to what extent the elements in any of the four domains form an intergral resource or medium which the stakeholder deploys. At the same time we can observe to what extent the stakeholder is able to analyse, monitor, and voice concerns

Table 3. IS failure types

Domain covered	*Development failures*	*Use failures*
Technical domain	technology problems	technical and operational problems
Data domain	view of data	data problems
User domain	view of user	conceptual problems; job satisfaction
Organizational domain	view of organization	organizational problems
Interference problems	goal problems; complexity problems	complexity of use; complexity of maintenance
Development process problems	communication problems; control problems; view of ISD process	n/a

about those elements. As stakeholders' interests can intersect with the four domains in either IS use or development, we can classify IS failures into use and development failures, giving 2×4 IS failure classes initially.

However, these eight classes are insufficient for developing a comprehensive framework, as problems may arise because the domains interact in the IS. These we call *domain interference* failures. Interference failures can take place within IS development or in IS use. Additionally, it is possible to note problems which have solely to do with stakeholders' concerns over the IS development process *vis-à-vis* organizational arrangements and processes. These we shall call *development process* failures. Thus, eleven principal failure classes can be distinguished.

Within these eleven principal failure classes it is possible to make further refinements—specifically in the development and interference failures—yielding a total of sixteen IS failure classes (see Table 3).

4.2 Development failures

Development failures include 'domain failures', i.e. problems in perceiving, analysing, designing, or configuring some domain; 'interference failures', i.e. problems in taking into account how domains interfere with each other; and 'development process failures', i.e. problems in carrying out and organizing ISD. We note nine IS failure classes which principally concern IS stakeholders' inability to design and implement the IS.

4.2.1 Domain failures

(*1*) *Technology problems.* Technological infrastructure and organizational policies may restrict the technology choices to inappropriate alternatives (Kling and Scacchi 1982). Or, stakeholders' expectations of the technology may be unrealistically high or low (Alter 1980); both, however, may lead to inadequate technological solutions. Occasionally, the technology used may depend on scarce resources like the ability to program in a particular programming language (Kling and Scacchi 1982). And rapid changes in information technology often bring risk and uncertainty to technology decisions.

(*2*) *View of data.* Systems development activities concentrate mainly on how data should be processed in the system. Less attention is paid to the social meaning and impact of data (Stamper 1985; Kensing 1984; Lyytinen 1987a). For this reason conflicting and different interpretations of data are seldom disclosed, leading to misinterpretations and harmful organizational action and decisions (Stamper 1985, Boland 1979).

(*3*) *View of user.* Systems development activities mainly concentrate on an idealized (average) user. They tend to neglect the cognitive, motivational, and skill variation in the user population. In addition, the dominating view of the user is highly mechanistic, denying learning and cognitive evolution with the system (Hedberg and Mumford 1975; Jarvinen 1986). Instead, the user must conform to the technical rigidity of the system interface imposed on him or her from the technical domain.

(4) *View of organization.* Systems development activities typically pay only scant attention to changes in the organizational domain. They tend to overlook the fact that organizational roles, job content, autonomy, work loads, and so on, can be radically affected by changes in the other IS domains. They also neglect increased dependency on the system, higher rigidity and red tape, and other harmful organizational impacts (cf. Senn 1979; Stamper 1985; Mayntz 1984).

Often the economic resources needed to develop the system are highly underestimated. Cost–benefit figures typically neglect important items such as the value of the existing system, the organizational costs of adopting the new system, and so on (Strassmann 1985). In consequence, most cost–benefit calculations are suspect, showing IS change in too positive a light. Many writers therefore claim that cost–benefit analyses serve legitimating purposes for planned decisions rather than for determining the best decision choice (Mayntz 1984). Kling (1980) also points out the opposite case where cost–benefit calculations were ignored and the decision was made on the basis of some ulterior motives.

4.2.2 *Interference failures*

(5) *Goal problems.* IS goals are ambiguous in most domains. They do not take into account dependencies between domains, focusing instead on quantitative aspects. They concentrate heavily on the technical domain and on the economic aspects of organizational performance. Moreover, the goals are quite often uncritically selected, assuming that computerization and the employment of mathematical (normative) models constitutes 'progress' (Senn 1979). Not surprisingly, goals usually express managements' and IS professionals' views about the IS.

(6) *Complexity problems.* In many cases the planned IS is highly complex, potentially exceeding the limits of human comprehension—particularly, that of certain stakeholder groups. In such cases, IS design deals with what Rosenberg (1980) calls 'totally incomprehensible systems'. There are two main sources of such complexity. The first originates from the inherent complexity of all four domains in general, and the technical domain in particular (Senko 1975; Rosenberg 1980). The second relates to the interactions among the four domains. These are much more subtle and harder to comprehend, with few theoretical constructs for dealing with them. It is not surprising, therefore, that most of the literature covers only the first type of complexity (Buckingham 1977; Rosen-

berg 1980). Yet, the second type of complexity is far more important in IS designs (cf. Lehman 1980), and cannot be adequately handled by technical fixes. IS stakeholders often find it difficult to understand the IS, to voice their concerns about suggested changes, and/or to mobilize action in time to defend their interests.

4.2.3 Development process failures

(7) *Communication failures.* The systems development process tends to be a specialist-dominated activity. IS professionals can initiate, control, and direct communication and decide when to end communications (Salaway 1984; Boland and Day 1982). Communication is problematic because of the unfamiliar languages employed by various stakeholders (participants) (Hammond 1974; DeBrabander and Edstrom 1977; DeBrabander and Thiers 1984). As a result, various IS stakeholders feel themselves alienated, their concerns not being sufficiently articulated. Other groups are seen as 'obtacles', preventing them from working properly.

(8) *Control failures.* Quality control in systems development is often insufficient. Control measures are limited and focus mainly on technical quality. Systems development tools and methods improve the work of IS professionals and strengthen their organizational status, but other stakeholder groups are not encouraged either to take part or to control the systems development process. The result is systems which typically do not fulfil their expectations (Ciborra and Bracchi 1983).

(9) *View of the IS design process.* The development process is typically viewed and understood as an overtly rational process which deploys available means to achieve stated economic ends. The process is structured as a linear problem-solving process, in a largely given problem space (Lanzara 1983)—an underlying theme of all life-cycle models. In addition, the process is viewed in Tayloristic terms: it is organized in a highly bureaucratic fashion emphasizing control and efficiency—a view that underlies most managerial ideologies of organizational action. This view largely ignores the real process of bargaining and conflict resolution that is typical in any organizational change, where various stakeholders participate, voice their interests, and mobilize action around the IS. In this sense, the largely rational image of systems development acts as a distorting filter of perception. It hinders developers (and methods) from properly 'understanding' that part of organizational reality which they deal with and change.

4.3 Use failures

Use failures fall into seven failure classes. These failures concern stakeholders' problems in realizing their expectations in IS use and maintenance.

4.3.1 Domain failures

(*10*) *Technical and operational problems.* Information systems are sometimes slow and unreliable. Their interfaces can be technically complex and awkward. Systems may have to be redesigned, and observed errors and bugs need to be fixed during system use. This increases IS maintenance costs and reduces the economic viability of the system. Alter (1980) regards these problems as largely transient, as technical problems can be circumvented by adequate redesigns. However, such a view may be too optimistic. As Kling and Scacchi (1982) point out, technical redesigns are sometimes difficult to accomplish owing to resource constraints, unstable computing arrangements, continuous shifts in computing technology, and the inherent complexity of software fixes.

(*11*) *Data problems.* Data in the system is often incorrect due to the lack of organizational controls, low motivation to the inputting of correct data, sabotage, or poor timing. Often it lacks relevance, because it is based on wrong classification schemes and measurement categories. Accordingly, data can distort the true 'picture' of the organization by conveying only selected parts of it. This may be ineffective or even dangerous (Nygaard and Handlykken 1981; Nygaard 1986; Boland 1979), because it may lead to inappropriate action. Data can also be incomprehensible so that it is misinterpreted or interpreted inconsistently, brought about by poor documentation, and/or focusing on the technical details of data (Alter 1980). Sometimes, data are simply missing, or if they exist, they are impossible to access in a cost-effective manner (Alter 1980; Senn 1979).

(*12*) *Conceptual problems.* Conceptual problems may result in errors of the 'third kind'—solving the wrong problem (Kilmann and Mitroff 1979; Mitroff 1980). As a consequence, systems address only those problems amenable to computer-based solutions (Bjorn-Andersen and Eason 1980). Inadequate conceptualization of the information system results in inflexible systems that cannot accommodate changes in the organization, user, or data domains. Wrong conceptualizations lead to ineffective and poor IS use.

(*13*) *Job satisfaction problems.* The use of an IS may deleteriously affect a user's job by degrading the work content, lowering the quality of the physical and social work environment, reducing control and autonomy, and/or inhibiting learning and cognitive evolution (Mumford 1983; Land and Hirschheim 1983).

(*14*) *Organization problems.* The information system may have organizationally undesirable properties. It may change organizational role patterns; decrease human contacts that are necessary for socialization and adoption of norms, values and organizational images; introduce red tape; and/or strengthen organizational rigidity. This alienates people and creates resistance among IS stakeholders. Sometimes, work processes and positions become obsolete, causing job losses—an aspect of considerable concern among some

IS stakeholders (Briefs *et al.* 1983). The IS also directly affects interest groups' power positions by changing access to organizationally critical data (Hedberg 1980; Kling 1980; Mayntz 1984; Markus 1983). Additionally, there are long-term impacts which may be latent for some time. In general, these deal with changes in job qualifications, their societal images, and shifts in the expert knowledge related to jobs (Goranzon 1983).

4.3.2 Interference failures

(*15*) *Complexity of use.* Information systems can be difficult to use and manage by end-users because of technical complexities, voluminous data descriptions, and complex organizational procedures related to IS use. This inherent complexity makes them error-prone and behave in unpredictable or even dangerous ways. There are a number of examples in which complex ISs were operated incorrectly, leading to fatalities (SEN 1983).

(*16*) *Complexity of maintenance.* Technical implementations of IS often involve errors that are hard to pin down and correct, and may lead to unpredictable system behaviour. The major reasons are: the sheer complexity and size of the software, and conceptual problems in understanding it properly. Locating errors is difficult, with even greater uncertainty in predicting the consequences of software change. In addition, the continuous evolution in the user, data, and organizational domains creates new demands for software changes (Lientz and Swanson 1980; Swanson 1984), which amounts to software maintenance that is extremely laborious and leading to accelerating maintenance costs.

4.4 Discussion

The failure types identified are not as distinct as portrayed. For example, development failures can have a dramatic impact on use failures (data problems, conceptual problems, organizational problems). In addition, one development or use failure class may be connected to several other failure classes of the same sort. For example, an insufficient and narrow view of data may lead to communication failures, or vice versa. ISs manifest a rich variety of problematic situations, and one type of IS failure can contribute to several other types.

It is important, therefore, to study not only *what* sorts of failures have occurred, but also *how* they are connected. In real life it is extremely difficult to distinguish between failures and causes. Instead, failures tend to form a cyclical network of elements connected by mutual causality (Boland 1979) where more encompassing impacts are gradually produced. It is important to observe that these cascading sequences are produced by the intentional reactions of stakeholders when faced with difficulties, and in this way they are purposefully produced. Due to this connectedness, failures gradually extend to cover a larger set of domains.

One consequence is that new forms of failures can be defined and reconstructed by analysing various failure histories. For example, the notions of 'interaction failure' or 'correspondence failure' could be reconstructed in multiple ways using other failure types and studying their connections. They could be viewed as systematic patterns of IS failure encountered by some IS stakeholders.

The lesson to be learnt is that the only way to grasp the features of IS failure is through a systems approach (Checkland 1981; Bignell and Fortune 1984), as failures constitute holistic, emergent situations with many interacting elements. The systems approach also recognizes the partiality and incompleteness of our understanding of any situation (Checkland 1981). Lanzara and Mathiassen (1985) call such systemic descriptions *maps*, and show that they necessarily involve interpretive cognitive constructs derived by symbolic interactions and communications. IS failure analysis is an interpretive activity directed at understanding IS problems and creating actions to resolve them.

The contents of mappings depend on participating stakeholder groups whose interests are seen to intersect with the problematic situation. For example, technical and operational problems affect both users and IS professionals. However, the corresponding maps of the situation are very different, as is the action space created to resolve the faced problems. For this reason, it is necessary to understand in what ways stakeholders relate to IS and how they build their maps of the situation.

In understanding IS failures, the systems approach also reminds us of the importance of the larger environment (cultural, organizational, economic, etc.) in which stakeholder groups act and can be located. These environments form the 'ecology' of the problematic situation and connect IS failure to the overall organizational context.

To summarize, any IS failure analysis is a complex, interpretive accomplishment, and involves the consideration of the following issues.

(1) What failure types are observed?
(2) How should failure type connections be studied?
(3) Who are the relevant stakeholders?
(4) How are the environmental aspects identified and included in the analysis?
(5) What failure concept criteria are used to decide on appropriate failure assessment?

It can be noted that by mapping the various assessment aspects together, a potential IS failure matrix of 2^{16} possible combinations of failure types is produced. For each cell we could additionally ask questions such as: whether the connections are studied or not, whether the environment is included in the analysis, which stakeholders (if any) are seen as relevant, and which sorts of failure concept criteria are being employed. Clearly, such a matrix is too large to be fully analysed in any real-life failure assessment. For this reason, one must consider which sort of failure assessment approach is appropriate for a given context. Fortunately, this task is eased by the fact that some of the cells are not reasonable alternatives, and thus the actual number of cells which need to be considered is much lower. However, the potential space is still great, and there may be a number of viewpoints to consider. It is also clear that many of the viewpoints have not been fully developed or employed in systems design.

In Table 4, we apply the five issues raised above to two well-known works on IS failures: Lucas's (1975) pioneering study on information system failures, and Markus and Robey's (1983) survey on the organizational validity of IS. The table shows that despite the extensive nature of both studies, the domains of the failure matrix

Table 4. Classification of Lucas's and Markus and Robey's works on IS failures

Aspect considered	*Lucas (1975)*	*Markus and Robey (1983)*
Failure types considered	technical & operational problems; data problems; job satisfaction; organizational problems; control problems; communication problems	conceptual problems; user satisfaction; organization problems
Failure connections studied	technical/job satisfaction; technical/organizational problems; communication problems/job satisfaction	none
Stakeholders covered	management; users; IS department	management users IS professionals
Environments covered	Situational/personal factors	Environment/system fit
Evaluation concept criteria	one dimension; formative; discrete; neutral; static; formal; *ex post/ex ante*; context independent	several; summative/formative; continuous; neutral; static; formal/informal; *ex post/ex ante*; context independent

covered by both is relatively small. To summarize: (1) they deal mostly with problems in the user and organizational domains; (2) only some connections between failure types are considered; (3) the division of stakeholders is coarse (and reflects the role prescriptions related to IS design); (4) environments of failures are covered superficially; and (5) the analyses focus on the rational side of evaluation. As these works cover a large portion of the IS failures literature, we believe that they indicate the general content and direction of IS failure research in the past.

5 Reasons for IS failure

IS failures are neither random nor spontaneous events. With the benefit of hindsight one can usually reconstruct a systematic pattern of events that led to the failure. The advantages of such a retrospective study are threefold. First, it is possible to provide some explanations as to why the IS failed. Second, these reconstructions can be used for learning, and hence in preventing them from happening again. Third, these formative evaluations may help in predicting new failures in the future. The formative study, however, adds a new dimension to IS failure assessment: it stresses the importance of studying those elements of the 'total system' that contributed to the failure. We term these elements the *reasons* for IS failure.

In light of the fact that failure types are connected and contribute to other failure types, one might ask: what is the distinction between IS failure types and IS failure reasons? The distinction is of course relative and depends on how boundaries of the situation are drawn; however, reasons and failure types are conceptually different. There are three criteria to distinguish between them. First, failure reasons are the factors and situations which lie outside the IS. They belong to the environment (as depicted in Fig. 3). Second, failure reasons are believed to have causal precedence over failures and are not assumed to be caused by any higher-order reasons (thus avoiding the problem of infinite regress). Third, failure reasons must have some 'causal' connection to failures, i.e. they must co-produce problems by conditioning or affecting behaviours and arrangements that produce the failures.

Unfortunately, there are few systematic and extensive studies on IS failure reasons. In addition, most of the works published on this subject contain relatively simple factor models (Lucas 1975, 1981; Keen and Scott-Morton 1978). These models try to reduce failure explanations to simple causal pairings between some factors, such as management support, user involvement, technical quality of the system, and the like. These models, therefore, only partially cover the reasons discussed in the literature. Moreover, their failure explanations are mechanistic—thus ignoring the role of actors' knowledgeable behaviour and their context.

Most of the IS literature explicitly ignores any consideration of failure reasons. Rather, the literature contains a wealth of ideas, methods, and suggestions about how to reduce the risk of IS failure by improving the state of the art in information systems development. Implicit in these considerations are assumptions about why systems have failed in the past. It is these implicit reasons—along with those explicitly stated—which are the basis for the set of reasons presented below[5].

The reasons for information systems failure can be divided into four major groups:

(1) features of the IS,
(2) features of the IS environment,
(3) features of the ISD process,
(4) features of the ISD environment.

'Features of the IS' cover those aspects that are typically thought of as part of the IS proper, i.e. the technical and data domains. 'Features of the IS environment' cover the other two domains: the user and organizational domains. 'Features of the ISD process' deal with recognizable aspects of the development process, methods, organization, and the like. 'Features of the ISD environment' cover the larger societal and cultural environment.

Each group can be further divided into several failure reason classes. We note twelve failure reason classes in total into which the existing IS literature can be classified (see Table 5). Although the boundaries between the twelve classes are not always clear-cut or easy to pin down, it is believed that the classes show distinct reasons for IS failure. In this way they can provide a foundation for a more systematic understanding of the mechanisms that produce various IS failures.

5.1 Features of the IS

This group of failure reasons sees failure as emanating from the information system itself. That is, there are features of the IS—typically hardware or software related—which cause it to fail. The literature adopting such a posture is considerable[6].

[5] It should be noted that the use of the term 'reason' does not presume (as is conventional) a strict causal relationship between some factors and a failure. In contrast, we subscribe to an interactionistic and hermeneutic perspective (Klein and Hirschheim 1983; Markus 1983), which pays attention to multiple, varying, and contextual interactions between various elements in the IS, and to understanding the actors' reasons and intentions in their actions around the IS. Accordingly, we do not presume that any straightforward causal explanation is possible. Instead, the study of reasons can at most provide a multi-linear, multi-causal description of the failure. Furthermore, IS failure has many backgrounds, and the interactions between these backgrounds may be complex and irreversible. Thus, the description can never be complete, and it cannot be used to predict in the same sense as natural science models. In other words, every failure description is a diagnostic mapping in the spirit discussed in the previous section.

[6] Although there is no shortage of literature exploring the features of the IS within the technical domain, the same cannot be said of the data domain. This appears to be an area which has largely been ignored by the IS community. Perhaps the only research which is being done that begins to address the data domain is the transaction cost work of Ciborra (1985, 1987).

Table 5. Classification of IS failure reasons

Group	Content	Type of reason
IS	(1) Technical and operational reasons —lack of sophisticated technology	mostly uncontrolled
IS Env	(2) Individual reasons —lack of fit of the IS to users' capabilities (cognitive style, stress adaptation, motivation)	mostly controlled
	(3) Organizational reasons —lack of fit of the IS to the rest of the organization (age, stage, context, etc.)	mostly uncontrolled
	(4) Environmental reasons —lack of fit of the IS to operating organization environment (stability of IS function, organizational incentives, etc.)	mostly uncontrolled
ISD	(5) Method-based reasons —lack of adequate and powerful methods	controllable
	(6) Decision-making-based reasons —lack of sufficient attention to types of decisions supported	controllable
	(7) Work-based reasons —lack of sufficient attention to nature of work	controllable
	(8) Contingency reasons —lack of sufficient attention to contingency factors in ISD (type of system, development environment, risks, etc.)	controllable
	(9) Implementation reasons —lack of sufficient attention to organizational implementation	controllable
	(10) System assumption-based reasons —insufficient attention to biased or wrong assumptions that drive ISD	controllable
ISD Env	(11) Analyst-based reasons —IS professionals' insufficient cognitive and social skills, and too limited behavioural codes	controllable
	(12) User-based reasons —users' insufficient skills and capabilities, and their limited knowledge of computing	controllable

(*1*) *Technical and operational reasons*. In general, this body of literature asserts that the primary cause of IS failure is the lack of sufficiently sophisticated technology. The available means to design, build, and configure the operational system in the technical domain are seen as inadequate. In consequence, systems are slow, unreliable, and lack functionality. Additionally, this situation has an adverse effect on the three other domains: (a) data is incorrect or in the wrong form, (b) users cannot understand or use the system, and (c) the result is user resistance in organizations (Markus 1983; Kling and Scacchi 1982). It is further argued that advances in technology will dramatically affect this course. There has been no shortage of proponents for this line of reasoning, nor has there been any shortage of recommended new technologies and tools. These include high-level languages, automated systems development environments (Teichroew *et al.* 1980; Nunamaker and Konsynski 1981), database technology (Deen 1977; Codd 1982) user-friendly interfaces (Shackel 1985), fourth-generation languages (Martin 1985), and fifth-generation computers (Moto-Oka 1981).

5.2 Features of the IS environment

This group can be divided into three classes:

(2) individual,
(3) organizational,
(4) environmental reasons.

(*2*) *Individual reasons*. This body of literature suggests that the key reason for IS failure stems from the lack of understanding about the individuals who will eventually use the system. The focus is on the user domain, claiming that IS failure is caused by an inadequate understanding of the human and behavioural issues, which results in a poor fit between how an individual processes information and how the IS adapts to it. Markus and Robey (1983) refer to this as a mismatch in the user–system 'fit'. Such a perspective would likely be adopted by organizational behaviourists or cognitive psychologists. The individual-based class of failure reasons has two dimensions. Dimension one is concerned with individual differences and with what the differences are. The literature on this topic is abundant, with various cognitive or individual-based reasons for the IS failure, such as: information processing styles and classes (Mason and Mitroff 1973; Mitroff and Mason 1983; Keen and Scott-Morton 1978; Huber 1983; Dickson *et al.* 1977), lateral asymmetry in brain function (Mintzberg 1976; Taggart and Robey 1980; Kerola *et al.* 1985); and cognitive limitations (Miller 1956; Shneiderman 1981). Dimension two is concerned with behavioural aspects, i.e. predicting what the real effects of introducing a new IS on the individual would be. In this area researchers have focused on the stress and anxiety of an individual (Argyris 1971a, b; Saunders 1974), motivation (Leavitt and Whisler 1958; Zmud 1979), expectations (Robey 1979, 1981), and so on.

(*3*) *Organizational reasons.* This body of literature suggests that ISs fail owing to the lack of consideration given to organizational issues. In other words, the main origin of IS failures lies in the organizational domain and how it interacts with the other domains. Such a perspective would be adopted by organizational theorists, and it draws upon the existing knowledge of organizations and organizational behaviour. The literature in this class has two separate dimensions. Dimension one focuses on organizational contextual variables: technology, structure, age, size, task uncertainty, and the like. Examples of the literature focusing on this dimension are: stage of an organization in its life cycle (Rubenstein *et al.* 1967), underlying technology of user departments (MacIntosh and Daft 1978), technology and environment (Olerup 1980), task uncertainty (Galbraith 1977), and so on. Dimension two reflects failure due to a lack of knowledge about how organizations work. Examples in this dimension are: 'real' organizational norms (Argyris 1971b), power shifts (Pettigrew 1973; Markus 1983), increased organizational rigidity and bureaucracy (Mowshowitz 1976; Bjorn-Andersen and Eason 1980), inappropriate structure of the IS function (Olson and Chervany 1980), rationality of the IS design (Argyris 1971a), and so on. These dimensions taken together are argued to determine the organizational 'fit' of the IS (Markus and Robey 1983; Ginzberg 1980).

(*4*) *Environmental reasons.* In this class, IS failure arises from the lack of consideration given to the larger organizational and functional environment into which the IS is brought. The environment is rather loosely defined to embrace those areas or concerns which surround the boundary between the IS and the external organization. This viewpoint is often held by a number of researchers in the IS area and management consultants. Some of the concerns have been: the stability of the IS function and the associated need for IS planning (King 1978; McFarlan and McKenney 1983; Sullivan 1985; McFarlan 1984; Earl 1987), knowledge and information explosion (Schwartz *et al.* 1980), lack of organizational incentives and controls (Kling 1977), implementation success factors (Ginzberg 1975; Keen and Scott-Morton 1978), the organization's environment and its fit with the system characteristics (Markus 1984; Robey 1981), organizational structural relationships between the data processing department and its users or within data processing itself (Ahituv and Neumann 1982).

5.3 Features of the ISD process

This group postulates that the primary reason for IS failure is the deficiency of the information systems development process. It asserts that some features of the ISD process create failures during development. This results in the IS being of poor quality with unwanted characteristics, causing the users to reject it. This category can be divided into six classes:

(5) methods,
(6) decision-making,
(7) nature of work,
(8) contingency,
(9) implementation,
(10) systems assumptions.

All six failure reason classes address the interactions between IS failure and the ISD process.

(5) *Methods reasons.* This class of failure reasons is the most thoroughly researched, and can easily be identified by the large amount of literature in the area. This perspective is very much the one adopted by information system researchers and computer scientists. It claims that the lack of rigorous and well-designed methods is the cause of IS failure. A method is seen as providing a developer with complete control of the IS and its impact on organizational performance. It recommends a list of guidelines on how to design a 'good' system (Lanzara 1983; Yeh *et al.* 1980; Couger *et al.* 1981; Olle *et al.* 1982, 1983, 1986).

The main concern in this class has been to provide methods to design and analyse the contents and functions of the IS in the technical and data domains. There are several 'schools of thought': functional requirements specifications (Ross and Schoman 1977; Gane and Sarson 1979; DeMarco 1978), data analysis (Davenport 1978; Palmer and Gradwell 1976; Olle *et al.* 1982), prototyping (Berrisford and Wetherbe 1979; Naumann and Jenkins 1982; Earl 1978; McCracken and Jackson 1981), and so on.

The concern in the development of methods is also reflected in the user and organizational domains. In the user domain, issues such as user dialogues and dialogue models can be noted (Shackel 1985). In the organizational domain, a family of methods referred to as the socio-technical methods (Bostrom and Heinen, 1977a, b; Mumford 1983; Land *et al.* 1980; DeMaio 1980) has been developed.

There are also a variety of methods which try to improve the characteristics of the ISD process. Most important among these have been suggestions regarding how to sequence development activities (Lehman 1984; Kerola 1980; Kerola and Freeman 1981; Lundeberg *et al.* 1981; Yourdon 1982). Others range from project control mechanisms (Keen 1982) and cost allocation schemes, to participative design methods (Mumford 1983; Hirschheim 1983, 1985a; Ives and Olson 1984), and project team composition techniques (White 1984). Recently, there has been interest in methods tht help to select other methods, referred to as 'contingency models' (Davis 1982; Ahituv and Neumann 1984).

(6) *Decision-making reasons.* This body of literature suggests that information systems have a tendency to fail because too little attention is paid to the types of decisions which are to be supported by the IS, i.e. interactions between the data, user, and organizational domains. This class contends that systems are too often designed to support decisions which are assumed to be highly structured, or will become so through the use of the system.

Numerous prescriptions have been offered by proponents of this view, which is largely populated by management scientists. Some of the most prominent examples are: inquiring systems (Mason and Mitroff 1973), better information requirements analysis (Bariff 1977; Taggart and Tharp 1977; Cooper and Swanson 1979), critical success factors (Rockart 1979; Henderson and West 1979), decision support systems (Keen and Scott-Morton 1978; Alter 1980; Sprague 1980); understanding the nature of measurement (Mason and Swanson 1981), and so on.

But as Mintzberg (1973) notes, most decisions within organizations are not highly structured. This applies even to decisions in the lower levels of the organizational hierarchy (Wynn 1979), even though it is at odds with the classical model of decision-making based on Anthony's work (Keen and Scott-Morton 1978). Moreover, numerous scholars have recently pointed out that many decisions do not follow a rational decision model (Huber 1981; Pettigrew 1973): information is not used simply for selection, but in after-the-fact justification and persuasion (Feldham and March 1981); most decisions and organizational activities are not individualistic in nature (Ciborra 1984, 1985), and hence many ISs do not support actual organizational decision-making. In consequence, many have claimed that systems assuming highly structured and individualistic decision-making environments are bound to fail.

(7) *Nature of work reasons.* This body of literature asserts that the primary cause of IS failure is the inappropriate attention paid to the users' (manager's) job, i.e. how the user and organizational domains interact. That is, systems are designed around what managers' jobs are thought to be, rather than what they actually are. The reality of managerial work is more unstructured and less deterministic than conventional wisdom seems to suggest. Some also contend that computer-based ISs have failed because they constrained the manager in the way he works—something which is not acceptable to the manager. Proponents of this view, mainly organizational scientists, are Leavitt and Whisler (1958), McLean (1979), Mintzberg (1973), Stewart (1971), and Hirschheim (1985b), among others.

(8) *Contingency reasons.* This is a diverse body of literature which claims that information systems have failed for a variety of simultaneously occurring reasons. There has been no particular cause which one can pinpoint as the major cause for failure in all situations—in every situation it is possible to discern a combination of reasons. Contingency writers suggest that there are several

dimensions which have to be looked at when designing and implementing information systems to prevent failure. Since no one approach can be used in all situations effectively, a contingency approach is needed.

The available contingency approaches differ from one another in many ways. In most cases they have been suggested by the information system community. Some of the concerns put forward by these approaches are: risk assessment (McFarlan 1981), contingency factors (Davis 1982; Iivari 1983; Ahituv and Neumann 1984; Naumann *et al.* 1980), and situation analysis (Schonberger 1980).

(*9*) *Implementation reasons.* This body of literature suggests that the key reason for IS failure is the lack of consideration given to the dynamic process of implementation. Failure is due to not understanding the dynamics of change in the user and organizational domains, which is different from that in the technical domain. Therefore, IS implementation should not be thought of as simply a point in time when the technical system goes into operation. The area of implementation research has attempted to address this issue for over a decade. Some of the points raised include: process and role prescriptions (Zmud and Cox 1979; Ackoff 1967; Argyris 1971a), dynamic nature of implementation (Ginzberg 1978a, 1979), a designer's role as a change agent (Ginzberg 1978b), and group dynamics (Highsmith 1978).

Part of the implementation literature contends that IS failure can often be associated with political 'games' people play to avoid using the IS. Keen (1981) refers to these games as 'counter-implementation strategies'. Whatever the name, these games are manifestations of interest-seeking strategies followed by certain IS stakeholders. This has resulted in several proposals for counter-counter strategies to be adopted by IS professionals to save the system. Some of these are: 'sell the system' (Alter 1980; Schewe *et al.* 1977), 'keep the system simple' (Alter 1980), 'hide complexity' (Alter and Ginzberg 1978), and so on.

(*10*) *Systems assumptions reasons.* This body of literature asserts that the major reason for IS failure is the erroneous assumptions about the data, user, and organization domains held by the developers—which are also built into tools and methods. These are assumptions—myths (Westerlund and Sjostrand 1978; Boland and Greenberger 1986) or metaphors (Lanzara 1983; Lyytinen and Lehtinen 1987)—about the design world. Such assumptions are commonly held to be valid, but have never been proven so. In fact, some of the best-known and widely accepted assumptions may well be fallacious. For example:

(a) systems have well defined and measurable organizational objectives (Huysmans 1970);
(b) systems serve individual decision-making (Ciborra 1984, 1985; Lehtinen and Lyytinen 1986; Lyytinen and Lehtinen 1987);

(c) users behave rationally (Keen and Scott-Morton 1978; Markus 1983);
(d) users want to use terminals (Keen 1976);
(e) users know the information they need (Ackoff 1967);
(f) systems include consistent information (Hedberg and Jonsson 1978; Ciborra 1984, 1985; Earl and Hopwood 1980);
(g) systems development is rational and well-structured (Ciborra and Bracchi 1983; Franz and Robey 1984; Markus 1984; Kling and Iacono 1984; Scacchi 1985);
(h) systems development is neutral and expert driven (Hedberg *et al.* 1976; Boland and Day 1982; Sandberg 1985);
(i) systems specification equals model building (Boland 1979; Goldkuhl and Lyytinen 1982);
(j) user participation increases system acceptability (Hirschheim 1983; Ives and Olson 1981; Powers and Dickson 1973).

5.4 Features of the ISD environment

In general, this group focuses on the broad cultural and societal environment of the ISD process, and finds certain features in it as reasons for IS failure. It is divided into two classes: systems-developer-based reasons and user-education-based reasons. They both address interactions between the ISD process and some part of the larger environment.

(*11*) *Systems developer reasons.* This body of literature contends that the primary reason for IS failure rests on insufficient and/or different cognitive capabilities, or behavioural codes for IS professionals. The literature in this class has three dimensions. Dimension one focuses on individual systems analysts' cognitive skills. Representative literature in this dimension comprises: studies on knowledge needed for systems success (Vitalari 1985), and the assessment of good systems analysts (Vitalari and Dickson 1983). Dimension two focuses on systems analysts' and users' different cognitive styles, and suggests that information systems would be more successful if users and system design professionals were more alike (Huysmans 1970; Hedberg and Mumford 1975; Kumar and Welke 1984). Dimension three focuses on the nature of the systems profession. That is, the systems profession has a series of norms and realities adhered to by the individuals who work within it, which inhibit successful ISD. The major problem with these norms is that they limit a systems analyst to concerns about technical matters and to pay only scant attention to human issues. Several examples in the literature are: the self-selection process and peer-orientations (Bjorn-Andersen and Hedberg 1977), historical reasons (Sackman 1971), held 'models of man' and value orientations (Hedberg and Mumford 1975; Kumar and Welke 1984), reward structure (Kling 1977), role of professional societies (Mumford and Pettigrew 1975; Klein and Lyytinen 1985; Klein and Hirschheim 1987), and IS professionals' attempts to reduce managerial control of their profession (Markus and Robey 1983).

(*12*) *User education reasons.* This body of literature insists that a critical factor in IS failure is that of user education. That is, users have not been educated in what to expect of computer-based information systems. They have expectations, which are, more often than not, unrealistic; nor have the users been educated in ISD. Solutions range from better educational methods and programmes, to approaches which allow users to learn by doing. Some of the suggestions offered in the literature include: top management education (Heany 1972), user education on systems and development (Lucas 1973; Bronsema and Keen 1984), prototypes as learning tools (Earl 1978; Alter 1980), scaling down user expectations (Dearden 1972; Ginzberg 1981), unlearning (Williams 1969; Podger 1979), educational ramification of systems development approaches (Hoyer 1979), and so on.

5.5 Discussion

The failure reason classes are not as distinct as portrayed. Several of them can occur concurrently, and in many cases there is a strong multi-collinearity between the factors. Therefore, in most cases the failure reason analysis must proceed by taking into account interactions between several failure reason classes and to study their dynamic properties. This makes the study of IS failure reasons a complex enterprise and further reduces the possibilities for statistical generalization.

It is not claimed that the presented classification is necessarily exhaustive and disjoint. Future studies may show that there are other failure reasons, or that the field can be classified in a better way. However, the novel classification presented here offers a 'road-map' to the existing literature and lets the entire terrain become visible. Moreover, by combining the failure reasons with the four domains of IS, a multitude of differences in failure reason analysis can be observed. Indeed, there are great differences with respect to the types of reasons that may cause IS failures.

In our survey, elements in all domains (except the data domain) were found to cause IS failures. Also the type, coverage, and depth of failure reason analysis was found to vary enormously. In some parts of the literature the reasons are offered without any in-depth empirical evidence and research. Other studies provide extensive statistical surveys to find significant correlations between speculated reasons and problematic situations. These differences can be accounted for largely by two factors. First, the number of disciplines drawn upon in failure studies is extensive; different academic disciplines (and even 'schools' within a discipline) have different standards of how to conduct 'good' research. This introduces pluralism into IS failure research, i.e. there are several ways to approach and study failure. The field, therefore, is in a state of dispersion rather than cohesion. Second, the failure studies are founded on different conceptions of what constitutes failure.

So far we have barely touched on how failure types relate to failure reasons. The sheer complexity of failure type assessment and the number of failure reason classes implies that any attempt to explain IS failures cannot cover all aspects. One must reduce the number of aspects considered, if any meaningful explanations are hoped for. Unfortunately, most of the literature on IS failures takes this too literally and excessively reduces and simplifies the situation by considering few measurable reason classes. Moreover, in many works the type of IS failure is left completely unexplored or discussed from a very narrow perspective.

The major rule followed in the literature seems to be that if failure reasons belong to a domain *X*, say the technical domain, then the primary failure types are found in the same domain, e.g. technical and operational problems. In addition, the studies then relate the other domains causally (or functionally) to observed failures in domain *X*. In this way one can derive the view that technical problems will create user satisfaction and organizational problems. Analyses usually proceed in just one direction, i.e. from the inner domains to outer domains, and from 'preconceived' reasons to failure types. The nature of indirection is usually causal and mechanistic (Kling and Scacchi 1982). In consequence, there are few studies on how failure types and failure reasons interact dynamically over longer periods of time, and where the interactions between domains are seen in many directions and governed by mutual causality.

6 Some implications for practice and research

A number of observations have been made on how IS failures have been studied in the past. In this section, the most important ones are restated, and their implications for practice and research are considered.

6.1 Implications for practice

Three implications are noted: multiplicity of IS failures, nature of approaches to cope with failures, and conditions that strengthen problem resolution. Our first observation was that IS failure is a multi-faceted phenomenon of immense complexity which defies any simple solution. It is therefore very unlikely that any IS is simply a success or a failure. Such statements made in practice usually suggest that a system is running and 'successful' in implementation terms. However, such claims usually hide all redistributive effects of IS design, and fail to appreciate the rich variety of stakeholders' expectations of the IS. Thus, the use of such terms is justified only from a very narrow perspective.

IS failure is not necessarily negative. It depends on the viewpoint one takes—there are always some IS stakeholders who gain from the failure. The crux of failure assessment is in understanding the direction and size of redistributive effects produced, and how various IS stakeholders organize their action around these effects. If all redistributive effects are avoided it is likely to lead to a very conservative development strategy. In this case information technology would be used simply to automate existing organizational procedures—not as a vehicle for organizational change. This can be counter-productive in the long run and can negatively affect the viability of the organization and interests of all stakeholders.

If redistributive effects are too radical, IS change can deeply disrupt the organizational situation. In such a case, it is important that any IS development initiative has strong political support from key stakeholder groups, and evidence of gains for each group. Unfortunately, many reported IS failures have been caused by insufficient support.

Finally, in some situations failure results simply from the inability to design and implement the system. In this case the ambitions are too high and do not match the technical (and organizational) skills available to design and implement the system.

Because no simple tactic is possible to prevent IS failure, more emphasis on understanding contextual features of IS is necessary. IS stakeholders should focus on those critical factors in the IS that will have adverse effects on their interests. IS failures should be studied as dynamic processes that can be shaped by stakeholders' action. This action depends on available interpretations and 'mappings' of the situation, i.e. how and in which light stakeholders make sense of the situation. The critical issue is the content and scope of the 'mapping' process—what are the conceptual means by which stakeholders inquire and reflect on IS.

Usually the mappings are done intuitively and their scope is very limited—in this sense they provide only fragmentary guidelines for actors' behaviour. Yet it seems likely that an improvement of the quality and scope of conceptual maps is crucial for better IS development.

We do not present a detailed account of how mappings could be improved (see Lanzara and Mathiassen 1985; Mason and Mitroff 1981), but to show the power and utility of the mapping concept, we outline some steps of the mapping procedure, using the concepts introduced in the previous sections. Every step in this process should be associated with continued surveillance, and supported by community tools that enhance problem identification, assessment, and bargaining (cf. Holt and Cashman 1981).

The first step in understanding any IS use or development process is to build up a stakeholder map. This should include all actors and actor groups whose interests and behaviour intersect with the IS. Here criteria outlined in subsection 2.3 can be used. It is also important to note that stakeholder mapping is a reflective activity—actors and actor groups must reason on their own position and associated interests.

The second step is to find out the problems faced by various stakeholder groups. This problem map includes all those features that call for action. The problems should include both existing problems and those anticipated through scenarios related to IS design. Problem analysis is inherently value-based, and thus requires the analysis of values (Kling 1985). The problems in the problem map should be related to different domains of IS activity and an associated failure type list can be used to check the completeness of the problem map. The problem map can be completed by the analysis of problem connections.

The third step is to link the problem map with the failure reasons. As reason analysis is never complete, the basic task is to analyse the most relevant reason classes for each observed problem area. This will yield a failure connection map for each problem area and reason class for each stakeholder group.

As problem areas are not necessarily shared among stakeholder groups and cannot be solved without conflict, the fourth step is to negotiate a sufficient political consensus on what measures are needed to resolve the problems. Basically, this requires clarification of responsibilities and creation of a mutually binding agenda of how to intervene in the problematic situation.

Clearly, failure mapping poses considerable difficulty and requires not only strong technical skills, but also great political and social competence (Markus 1984; Vitalari 1985; Newman and Rosenberg 1985). In particular, it demands the capacity to anticipate changes in the political climate and skills to initiate and maintain social change. One problem, of course, is how to achieve such skills. It is unlikely that they can simply be learned from standard textbooks in systems design or through formal training. Good candidates for improving these competences are role-playing and hermeneutic forecasting (Klein and Hirschheim 1983), in-house analysis and study of failure histories (Walton and Vittori 1983), quality circles (Thompson 1982), and so on.

Another condition for preventing IS failure is to establish organizational arrangements that help to channel stakeholders' influences during IS design so as to let them meet one another. Here, more open, democratic, and participatory arrangements seem to be necessary. A negative consequence of this may be a slower development pace, but the reward is higher fulfilment of expectations, reduced resistance, and more informed and sophisticated IS clientele (Hirschheim 1983, 1985a).

6.2 Implications for IS research

Several research implications emerge from the proposed framework, both of an empirical and theoretical nature. In particular, the research implications deal with overcoming some of the weaknesses of studying IS failures in the past. In the following, five weaknesses are identified and discussed. These are: inadequate understanding of the nature of IS failure, inadequate attention to stakeholder analysis, inadequate clarification of reasons for IS failure, too simple explanatory models, and too limited research methods. Some suggestions are offered about the type of research needed in the future.

6.2.1 Inadequate identification of the type of failure

The expectation failure notion implies a clear understanding of the types of problems met by a particular stakeholder group. Unfortunately, the literature has not been very detailed in exploring how information systems have failed. IS researchers often suggest too general and vague descriptions of failure histories. For example, Lucas (1975, pp. 2–3) enumerates several observed problems without a sufficiently detailed clarification of their differing nature:

> 'users do not understand much of the output they receive (data problem), there is a duplication of input and output (data problem) . . . and changes are frequently made without consulting users (process problem) . . . users complain of information overload (conceptual problem) . . . there are also complaints about difficulty of obtaining changes in existing systems (process problem/complexity of maintenance).'

More refined analyses of IS development and use problems have been undertaken in recent years (Markus 1983; Kling and Scacchi 1982; Kling and Iacono 1984; Franz and Robey 1984; Alter 1980). However, despite these advances the field still lacks a more comprehensive taxonomy of IS failures. We believe that such a taxonomy could ease the comparison of research results, help to generalize research findings, and to focus research resources on a more informed basis. Some initial steps in this direction have been taken in this paper.

6.2.2 Inadequate identification of IS stakeholders

The expectation failure notion stresses the importance of identifying those stakeholders who perceive the IS has failed. In contrast, the literature has not been very detailed in identifying to whom the failure has occurred. For example, in Lucas's study (1975) user problems were generalized to concern all IS stakeholder groups; he assumed that all end-user groups had common interests. But what are the justifications for such an assumption?—none was provided.

We need more studies on how various stakeholder groups are affected by IS development and use. Most studies which adequately identify different IS stakeholders are of the case study variety (Markus 1983; Kling and Scacchi 1982; Kling and Iacono 1984).

More extensive statistical surveys (similar to Lucas's (1975) pioneering work) in which stakeholders' interests are sufficiently observed, are virtually non-existent. There is a lack of a comprehensive list of criteria for identifying relevant stakeholders, although works by Ginzberg and Zmud (1986) and Steinmuller (1984) are steps in the right direction. There is also a paucity of methods by which to determine and formulate their expectations adequately (Ciborra and Bracchi 1983).

6.2.3 *Inadequate identification of reasons for failure*

The expectation failure notion suggests that a more detailed understanding of the environment in which stakeholders act is required for clarifying why ISs fail. This would provide a more exact and comprehensive account of the 'ecology' of IS failure. Again, the IS literature is too superficial on this subject. It is almost impossible to classify the empirical research in terms of which reasons and failure types are covered by different studies.

Most of the literature is also highly speculative and too general. For example, most improvements suggested in new design methods and tools assume that IS failures are caused by the lack of powerful methods (Olle *et al.* 1982). In a similar vein, implementation studies presume that IS failures are caused by the lack of attention paid to the implementation process. When failure reasons are considered more thoroughly, several weaknesses can typically be found. First, reasons covered by the study include only those which are easy to quantify and subject to the rigours of statistical tests. For example, in Lucas's study (1975) four reasons were considered: (1) technical and operational reasons, (2) organizational reasons, (3) individual reasons, and (4) environmental reasons. All other reasons were simply ignored. However, there is no reason to believe that his assumptions are correct, not even that the reasons included in his model are necessarily the most important ones (cf. Keen and Scott-Morton 1978). In addition, most of the published studies insufficiently distinguish different types of IS failure.

Recent studies are more careful in identifying various reasons and their relationships to different failure types (Markus 1983; Kling and Scacchi 1982). However, a more systematic account of these relationships is clearly needed. In Table 6, we offer a preliminary classification of which reasons are relevant for producing different types of IS failures. Two relationships are distinguished: relevant (R) implies that there is a strong correlation between the reason type and the failure type; partly relevant (P) means that there are situations where the connection can exist.

6.2.4 *Inadequate models of IS failure*

The expectation failure notion suggests that one needs a detailed understanding of the mechanisms by which various elements

Table 6. IS failure type and reason connection matrix

Failure types	*Failure reasons*											
	1	2	3	4	5	6	7	8	9	10	11	12
Development failures												
Technology problems	R	P	—	—	R	—	—	—	—	—	—	—
View of data	P	P	P	P	R	—	—	P	—	P	P	—
View of user	—	R	—	P	P	P	P	P	P	P	P	—
View of organization	—	P	R	R	R	R	R	R	R	R	R	P
Goal problems	—	P	R	R	P	R	R	P	P	R	P	P
Complexity problems	R	P	P	P	P	P	P	—	P	P	—	—
Communication problems	—	—	—	—	R	R	R	—	R	R	R	R
Control problems	—	—	—	—	R	R	R	P	R	P	R	R
View of ISD	—	—	—	P	P	P	P	P	P	R	R	R
Use failures												
Technical problems	R	—	—	—	R	—	—	—	—	—	—	—
Data problems	P	P	P	P	R	R	P	—	—	P	—	—
Conceptual problems	—	R	P	—	R	R	R	P	—	P	—	R
User satisfaction	P	R	R	—	R	P	P	P	R	P	P	R
Organizational problems	—	—	R	R	R	P	P	P	R	P	P	P
Complexity of use	P	R	—	—	P	P	P	—	P	—	—	R
Complexity of maintenance	R	—	P	P	R	—	—	—	—	—	—	—

Key: P = partly relevant;
R = relevant

interact to produce problematic situations. Explications of such mechanisms are called failure models.

The identification of several stakeholder groups, the environments into which they interact, and the recognition of the active role of various stakeholders in producing IS failures complicates the content of models. The expectation failure concept suggests using multi-causal, hermeneutic models instead of the causal models advocated in the mainstream of the IS literature (Klein and Hirschheim 1983; Markus and Robey 1986).

This also implies that the use of single pairings (cause–effect chains) between classes of reasons and classes of failures in explaining IS failure is too simplistic and neglects the dynamic nature of failure. Instead, it is believed that there are several factors that simultaneously and/or at different points of time co-produce the failure pattern. Hence, IS failure cannot be prevented by manipulating just one or two factors in a stakeholders' environment, even though most of the IS literature in the past has advocated such prescriptions (and concentrated on directly controllable reasons). The causal models also tend to ignore the complex and important role of social arrangements ('webs' according to Kling and Scacchi 1982) which enable stakeholder groups to meet and to channel their influence in reaction to demands created by the IS. Thus, it is not surprising that the IS research literature has been poor in explaining differences in failure rates, e.g. by varying organizational settings and arrangements, cultural differences, social mechanisms and strategies supporting stakeholders' action, and so forth.

6.2.5 *Limited research methods*

Empirically studying the complex interactions associated with IS failure requires the application of new research methods. In Table 6, only some of the pairings can be explored with traditional research methods: empirical surveys, field studies, and laboratory experiments (Lucas 1975; Jenkins 1985). Some of the pairings—for example, relationships between systems-developer-based issues and process characteristics—can be studied by longitudinal research methods (Vitalari 1985). Still others, such as the impact of implementation process on organizational problems or the impact of methods on solving conceptual problems, may require the adoption of more qualitative research methods such as action research and ethnography.

There are also important research implications on how to use the proposed framework. It can be used to propose new empirical research problems, to identify unresearched areas, or to make comparative analyses between different research results. This suggests further research in several areas. Among these we note the following.

(*a*) *More extensive case studies.* We need to explore in more detail what problems stakeholder groups have, how they are co-produced by their activities, how they inquire into these problems, and how they arrange their actions around them. In addition, more studies are needed to explore how groups reason about problems and what reasons they find for these problems. This would suggest cultural comparisons and the clarification of how technological, cultural, social and political environments interact in failure (Klein and Hirschheim 1987).

(*b*) *More extensive statistical analyses.* We do not have reliable statistical data on which problems are typical for different stakeholder groups and in which environments. Such statistical data would supplement more qualitative studies and provide indicators on which types of problems are frequent. This would help in allocating research resources and to co-ordinate alternative research approaches.

(*c*) *More theoretical work on taxonomies.* We need a more comprehensive taxonomy of IS failures and the reasons for failure. This requires that a more complete theory of the 'ecology' of IS be developed, taking into account: the type of IS, its organizational purpose, organizational function served, existing organizational structure, the methods and tools used, types of IS technology employed, arrangements of the IS function, and so on (cf. Boland and Hirschheim 1987).

(*d*) *More theoretical work on failure models.* We need more theoretical studies on mechanisms that produce IS failures. This requires that ISs be studied in institutionalized settings. IS failure studies should be related more closely to sociological and organizational theories of organizational action: for example, Giddens's (1984) theory of structuration. Giddens's approach observes both the structural (objective) and process (subjective) aspects of social action and combines them in a theoretical account of social activity. One important parameter of Giddens's theory is the concept of institutionalization, i.e. how social activity is constrained by social mechanisms. This aspect has been largely ignored in the IS failure studies in the past. Another possibility is the application of Habermas's (1984) theory of communicative action. Initial attempts at applying Habermas's ideas within the domain of IS can be found in Lyytinen (1986), Klein (1986), and Lyytinen *et al.* (1986).

(*e*) *More studies on how to use failure models prescriptively*. The IS failure framework presented here could be used to diagnose IS failure risks and to plan stakeholders' action. In a research setting, researchers and practitioners could use the expectation failure concept to identify IS problem types for each stakeholder group. Measures that might reduce IS failure risks could accordingly be developed. It is, however, important to note that only some of the failure risks can be directly affected by carefully chosen implementation and development strategies (Hirschheim *et al.* 1984). Several failures are caused by cultural, structural, and societal constraints which are very difficult to change in the short run.

(*f*) *Related conceptual work*. It should by now be clear that one's perception of failure is strongly conditioned by the conceptual views one holds about the IS and its development. This suggests that more research is needed to classify and categorize conceptual views (myths) that make a specific IS failure a significant phenomenon. Moreover, because IS failure is associated with conceptions of how one ought to behave in using and developing ISs, an important research concern is the ethics of IS failure, and how ethical issues are related to the notion of IS failure (Kling 1978; Klein 1984).

7 Summary

IS failure is a complex issue which defies simple explanations. Because of the great amount of both conceptual and practical confusion surrounding IS failure, we have sought to highlight the key anomalies which exist in the literature. To this end, a classification framework has been proposed which allows the numerous aspects of failure to be considered. The framework also suggests where our knowledge is limited and where further research is needed. It is our belief that IS failure is a legitimate research area, one where considerable effort has so far been expended, but with little direct reward. We hope that our survey and classification of the literature will help to direct further work, and that future research will ultimately result in the demise of information systems failures.

References

Ackoff, R. L. (1967). Management misinformation systems. *Management Science* **14** (4), B147–56.

Ahituv, N. and Neumann, S. (1982). *Principles of information systems for management*. Wm. C. Brown, Dubuque, Iowa.

Ahituv, N. and Neumann, S. (1984). A flexible approach to information system development. *MIS Quarterly* **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 Management Review* **19**, 23–31.

Argyris, C. (1971a). Resistance to rational management. *Innovation*, No. 10.

Argyris, C. (1971b). Management information systems: the challenge to rationality and emotionality. *Management Science* **17** (6), 275–82.

Bailey, J. and Pearson, S. (1983). A tool for computer user satisfaction. *Management Science* **29** (5), 530–45.

Bakopoulos, J. Y. (1985). Toward a more precise concept of information technology. In Gallegos, L., Welke, R., and Wetherbe, J. (eds.) *Proceedings of the 6th International Conference on Information Systems*, pp. 17–24. Indianapolis.

Bariff, M. (1977). Information requirements analysis: a methodological review. Decision Sciences Dept., Un. of Pennsylvania, Working paper 76-08-2, January.

Berrisford, T. and Wetherbe, J. (1979). Heuristic development: a redesign of systems design. *MIS Quarterly* **2** (1), March.

Bignell, S. and Fortune, J. (1984). *System failures*. Manchester University Press, Manchester.

Bjorn-Andersen, N. and Eason, K. (1980). Myths and realities of information systems contributing to organizational rationality. In Mowshowitz, A. (ed.) *Human choice and computers-2*, pp. 97–109. North-Holland, Amsterdam.

Bjorn-Andersen, N. and Hedberg, B. (1977). Designing systems in an organizational perspective. In Nystrom, P. and Starbuck, W. (eds.) *Prescriptive models of organizations*. TIMS Studies in the Management Sciences, Vol. 5.

Bjorn-Andersen, N. and Pedersen, P. (1980). Computer facilitated changes in the management power structure. *Accounting, Organizations and Society* **4** (2).

Blackler, F. and Brown, C. (1986). Theory and practice of evaluation: the case of the new information technologies. In Davis, G. and Bjorn-Andersen, N. (eds.) *Information systems assessment*. North-Holland, Amsterdam.

Boehm, B. (1981). *Software economics*. Prentice-Hall, Englewood Cliffs, NJ.

Boland, R. J. (1979). Control, causality and information system requirements. *Accounting, Organizations and Society* **4** (5), 259–72.

Boland, R. J. and Day, W. (1982). The process of system design: a phenomenological approach. In Ginzberg, M. and Ross, C. (eds.) *Proceedings of the Third International Conference on Information Systems*, pp. 31–45. Ann Arbor, Michigan.

Boland, R. J. and Greenberger, R. (1986). Metaphorical structuring in problem formulation: an exploratory study. In Thomas, H. and Pondy, L. (eds.) *Managing Ambiguity and Change*. John Wiley & Sons, Chichester.

Boland, R. J. and Hirschheim, R. A. (eds.) (1987). *Critical issues in information systems research*. John Wiley & Sons, Chichester.

Boland, R. J. and Pondy, L. (1985). Accounting in organizations: the union of natural and rational perspectives. BEPR Faculty Working Paper No. 811, College of Commerce and Business Administration, University of Illinois, Urbana-Champaign.

Bostrom, R. B. and Heinen, J. S. (1977a). MIS problems and failures: socio-technical perspective—Part I: the causes. *MIS Quarterly* **1** (3), 17–32.

Bostrom, R. P. and Heinen, J. S. (1977b). MIS problems and failures: a socio-technical perspective—Part II: the application of socio-technical theory. *MIS Quarterly* **1** (4), pp. 11–28.

Briefs, U., Ciborra, C. and Schneider, L. (eds.) (1983). *Systems design, for, with and by the users*. North-Holland, Amsterdam.

Bronsema, G. and Keen, P. (1984). Education intervention and implementation in MIS. *Sloan Management Review*, Summer, pp. 35–43.

Brooks, F. (1974). *The mythical man-month*. Addison-Wesley, Reading, Mass.

Bubenko, J. A. (1986). Information systems methodologies—a research view. SYSLAB Report No. 40, The Systems Development and Artificial Intelligence Laboratory, University of Stockholm, Sweden.

Buckingham, R. A. (ed.) (1977). *Education and large information systems*. North-Holland, Amsterdam.

Canning, R. G. (1977). Getting the requirements right. *EDP Analyzer* **15** (7), 1–14.

Checkland, P. (1981). *Systems thinking, systems practice*. John Wiley & Sons, Chichester.

Ciborra, C. (1984). Information systems and organizational exchange: a new design approach. In Bemelmans, T. (ed.) *Beyond productivity—information systems for organizational effectiveness*, pp. 135–45. North-Holland, Amsterdam.

Ciborra, C. (1985). Reframing the role of computers in organizations. In Gallegos, L., Welke, R., and Wetherbe, J. (eds.) *Proceedings of the Sixth International Conference on Information Systems*, pp. 57–69. Indianapolis.

Ciborra, C. (1987). Research agenda for a transaction cost approach. In Boland, R. and Hirschheim, R. (eds.) *Critical issues in information systems research*, pp. 253–74. John Wiley & Sons, Chichester.

Ciborra, C. and Bracchi, G. (1983). Systems development and auditing in turbulent contexts: towards a new participative approach. In Wysong, E. M. and de Lotto, I. (eds.) *Information systems auditing*, pp. 41–52. North-Holland, Amsterdam.

Codd, E. F. (1982). Relational database: a practical foundation for productivity. *Communications of the ACM* **25** (2), 109–17.

Conrath, D. and duRoure, G. (1978). Organizational implications of comprehensive communication information systems—some conjectures. Working Paper, Institute d'Administration des Enterprises, Centre d'Etude et de Recherche sur les Organizations et la Gestion, Aix-en-Provence.

Cooper, R. B. and Swanson, E. B. (1979). Management information requirements assessment: the state of the art. *Data Base* **11** (2), 5–16.

Couger, J. D., Colter, M., and Knapp, R. (eds.) (1981). *Advanced systems development/feasibility techniques*. John Wiley & Sons, New York.

Cronan, T. P. (1984). Application systems development: a communication model for business users and DP personnel. *Data Base* **15** (4), 21–6.

Davenport, R. (1978). Data analysis for database design. *Australian Computer Journal* **10** (7).

Davis, G. (1982). Strategies for information requirements determination. *IBM Systems Journal* **21** (1), 4–30.

Davis, G. and Olson, M. (1985). *Management information systems: conceptual foundations, methods and development*. McGraw-Hill, New York.

Dearden, J. (1972). MIS is a mirage. *Harvard Business Review* **3** (2), 90–9.

DeBrabander, B. and Edstrom, A. (1977). Successful information system development projects. *Management Science* **24** (2), 191–9.

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. *Management Science* **30** (2), 137–55.

Deen, S. (1977). *An introduction to database management systems*. MacMillan, London.

DeMaio, A. (1980). Socio-technical methods for information systems design. In Lucas, H., Land, F., Lincoln, T., and Supper, K. (eds.) *The information systems environment*, pp. 105–22. North-Holland, Amsterdam.

DeMarco, T. (1978). *Structured analysis and system specification*. Yourdon Press, New York.

Dickson, G. W., Senn, J. A., and Chervany, N. L. (1977). Research in management information systems: the Minnesota experiments. *Management Science* **23** (9), 913–23.

Earl, M. (1978). Prototype systems for accounting information and control. *Accounting, Organizations and Society* 3 (2).

Earl, M. (1987). Information systems strategy formulation. In Boland, R. and Hirschheim, R. (eds.) *Critical issues in information systems research*, pp. 157–78. John Wiley & Sons, Chichester.

Earl, M. and Hopwood, A. (1980). From management information to information management. In Lucas, H., Land, F., Lincoln, T., and Supper, K. (eds.) *The information systems environment*, pp. 3–12. North-Holland, Amsterdam.

Ein-Dor, P. and Segev, E. (1978). Organizational context and the success of management information systems. *Management Science* **24** (10), 1064–77.

Etzioni, A. (1967). *The active society: theory of societal and political processes*. The Free Press, New York.

Feldham, M. S. and March, J. G. (1981). Information in organizations as signal and symbol. *Administrative Science Quarterly* **26**, 171–86.

Franz, C. R. and Robey, D. (1984). An investigation of user-led system design: rational and political perspectives. *Communications of the ACM* **27** (12), 1202–9.

Galbraith, J. (1977). *Organization design*. Addison-Wesley, Reading, Mass.

Gane, C. and Sarson, T. (1979). *Structured systems analysis: tools and techniques*. Prentice-Hall, Englewood-Cliffs, NJ.

Giddens, A. (1984). *The constitution of society: outline for a theory of structuration*. Polity Press, Cambridge.

Ginzberg, M. J. (1975). A process approach to management science implementation. Ph.D. Thesis, MIT, Boston.

Ginzberg, M. J. (1978a). Steps towards more effective implementation of MS and MIS. *Interfaces* **8** (3), 259–300.

Ginzberg, M. J. (1978b). Redesign of managerial tasks: a requisite for successful decision support systems. *MIS Quarterly* **2**.

Ginzberg, M. J. (1979). *A study of the implementation process*. TIMS Studies in Management Sciences **13**.

Ginzberg, M. J. (1980). An organizational contingencies view of accounting and information systems implementation. *Accounting, Organizations and Society* **5** (4), 369–82.

Ginzberg, M. J. (1981). Early diagnosis of MIS implementation failure—promising results and unanswered questions. *Management Science* **28** (7), 459–78.

Ginzberg, M. J. and Zmud, R. (1986). Evolving criteria for information systems assesment. In Davis, G. B. and Bjorn-Andersen, N. (eds.) *Information systems assessment*, North-Holland, Amsterdam.

Gladden, G. R. (1982). Stop the life-cycle, I want to get off. *Software Engineering Notes* **7** (2), 35–9.

Goldkuhl, G. and Lyytinen, K. (1982). A language action view of information systems. In Ginzberg, M.

and Ross, C. (eds.) *Proceedings of the 3rd International Conference on Information Systems*, pp. 13–30. Ann Arbor, Mich.

Goranzon, B. (ed.) (1983). *Datautvecklingens filosofi, tyst kunskap och ny teknik*. Carlson & Jonsson, Stockholm (in Swedish).

Habermas, J. (1984). *The theory of communicative action—reason and the rationalization of society*, vol. 1. Beacon Press, Boston.

Hamilton, S. (1981). Evaluation of information system effectiveness: a comparison of evaluation approaches and evaluation viewpoints. MISRC-WP-81-03, Graduate School of Business Administration, University of Minnesota.

Hammond, J. (1974). The roles of the manager and management scientist in successful implementation. *Sloan Management Review* **16**.

Heany, D. (1972). Education: the critical link in getting managers to use management systems. *Interfaces* **2** (3).

Hedberg, B. (1980). Using computerized information systems to design better organizations and jobs. In Bjorn-Andersen, N. (ed.) *Human side of information processing*, pp. 19–33. North-Holland, Amsterdam.

Hedberg, B. and Jonsson, S. (1978). Designing semi-confusing information systems for organizations in changing environments. *Accounting, Organizations and Society* **3** (1), 47–64.

Hedberg, B. and Mumford, E. (1975). The design of computer systems: man's vision of man as an integral part of the system design process. In Mumford, E. and Sackman, H. (eds.) *Human choice and computers*, pp. 31–59. North-Holland, Amsterdam.

Hedberg, B., Nystrom, P., and Starbuck, W. (1976). Camping on see-saws: prescriptions for self-designing organizations. *Administrative Science Quarterly* **24**, 40–63.

Henderson, J. and West, J. (1979). Planning for MIS: a decision-oriented approach. *MIS Quarterly* **3** (2).

Highsmith, J. (1978). Solving design problems more effectively. *MIS Quarterly* **2** (1).

Hirschheim, R. A. (1983). Assessing participative system design: some conclusions from an exploratory study. *Information & Management* **6** (1), 317–27.

Hirschheim, R. A. (1985a). User experiences with and assessment of participative systems design. *MIS Quarterly* **9** (4), 295–303.

Hirschheim, R. A. (1985b). *Office automation: a social and organizational perspective*. John Wiley & Sons, Chichester.

Hirschheim, R. A. (1986). The effect of a priori views on the social implications of computing: the case of office automation. *ACM Computing Surveys* **18** (2), 165–95.

Hirschheim, R. A. and Smithson, S. (1986). A critical analysis of information systems evaluation. In Davis, G. B. and Bjorn-Andersen, N. (eds.) *Information systems assessment*. North-Holland, Amsterdam.

Hirschheim, R. A., Land, F. F., and Smithson, S. (1984). Implementing computer-based information systems in organizations—issues and strategies. In *Proceedings of Interact-84, 4–6 September 1984*, pp. 356–64. London.

Holt, A. and Cashman, P. (1981). Designing systems to support cooperative activity: an example from software maintenance management. In *Proceedings of IEEE Compsac 81, Chicago, Illinois, November*.

Hoyer, R. (1979). Pedagogiska krav pa systemutvecklings-modeller. In Nurminen, M. and Jarvi, T. (eds.) *Proceedings of the Second Scandinavian Research Seminar on Systemeering Models, Tietojenkasittelyliitto, Helsinki* (in Swedish).

Huber, G. P. (1981). Organizational science contributions to the design of decision support systems. In Fisk, G. and Sprague, R. H. (eds.) *Decision support systems: issues and challenges*, pp. 45–56. Pergamon Press, Oxford.

Huber, G. P. (1983). Cognitive style as a basis for MIS and DSS designs: much ado about nothing? *Management Science* **29** (5), 567–77.

Huysmans, J. (1970). The effectiveness of the cognitive style constraint in implementing operations research proposals. *Management Science* **17**.

Iivari, J. (1983). Contributions to the theoretical foundations of systemeering research and the PIOCO-model. Ph.D. thesis, Acta Universitatis Ouluensis, Ser A 150, Institute of Data Processing, University of Oulu, Oulu.

Ives, B. and Olson, M. (1981). User involvement in information systems: a critical review of the empirical literature. CRIS 15, GBA 81-07, New York University, New York.

Ives, B. and Olson, M. (1984). User involvement and MIS success: a review of research. *Management Science* **30** (5), pp. 586–603.

Jarvinen, P. (1986). Flexibility of software as a dimension of information systems assessment. In Davis, G. and Bjorn-Andersen, N. (eds.) *Information systems assessment*. North-Holland, Amsterdam.

Jenkins, M. (1985). Research methodologies and MIS research. In Mumford, E., Hirschheim, R., Fitzgerald, G., and Wood-Harper, A. T. (eds.) pp. 103–17. North-Holland, Amsterdam.

Jonsson, S. (1986). New Technology and old management accounting—an organizational process view. Paper presented of the Annual Meeting of the American Accounting Association, New York, 20–22 August.

Keen, J. S. (1982). *Managing software development*. John Wiley, Chichester.

Keen, P. (1975). Computer-based decision aids: the evaluation problem. *Sloan Management Review* **16** (3).

Keen, P. (1976). Interactive computer systems for managers: a modest proposal. *Sloan Management Review* **17** (2).

Keen, P. (1981). Information systems and organizational change. *Communications of the ACM* **24** (1), 24–33.

Keen, P. and Scott-Morton, M. (1978). *Decision support systems: an organizational perspective*. Addison-Wesley, Reading, Mass.

Keider, S. (1978). Why MIS projects fail. *Datamation*, December.

Kensing, F. (1984). Towards evaluation of methods for property determination—a framework and critique of Yourdon–DeMarco approach. In Bemelmans, T. (ed.) *Beyond productivity: information systems development for organizational effectiveness*, pp. 325–38. North-Holland, Amsterdam.

Kerola, P. (1980). On infological research into the systemeering process. In Lucas, H., Land, F., Lincoln, T., and Supper, K. (eds.) *The information systems environment*, pp. 187–98. North-Holland, Amsterdam.

Kerola, P. and Freeman, P. (1981). A comparison of lifecycle models. In *Proceedings of the 5th International Conference on Software Engineering, San Diego*.

Kerola, P., Weckroth, J., Keinanen, P., Komulainen, S., Nuutinen, R., Pankkonen, K., Simila, J., and Tahvaninen, A. (1985). Research on human-centered information systems use and development—final report. Research Papers Series A6, Institute of Data Processing, University of Oulu, Oulu (in Finnish).

Kilmann, R. and Mitroff, I. (1979). Problem defining and the consulting-intervention process. *California Management Review*, Spring.

King, W. (1978). Strategic planning for management information systems. *MIS Quarterly* **2** (1), 27–37.

King, W. and Rodriguez, J. (1978). Evaluating management information systems. *MIS Quarterly* **2** (3).

King, W. and Rodriguez, J. (1981). Participative design of strategic decision support systems. *Management Science* **27** (6), 717–26.

Klein, H. (1984). Which epistemologies for future systems research. In Saaksjarvi, M. (ed.) *Report of the 7th Scandinavian Research Seminar on Systemeering (Part I)*, pp. 60–90. Helsinki Business School, Helsinki.

Klein, H. (1986). Organizational implications of office systems: toward a critical social action perspective. In Verrijn-Stuart, A. and Hirschheim, R. (eds.) *Office systems*, pp. 143–59. North-Holland, Amsterdam.

Klein, H. and Hirschheim, R. A. (1983). Issues and approaches to appraising technological change in the ofifce: a consequentalist perspective. *Office: Technology & People* **2** (1), 15–42.

Klein, H. and Hirschheim, R. A. (1985). Fundamental issues of decision support systems: a consequentalist perspective. *Decision Support Systems* **1** (1), 5–23.

Klein, H. and Hirschheim, R. A. (1987). Social change and the future of information systems development. In Boland, R. J. and Hirschheim, R. A. (eds.) *Critical issues in information systems research*, pp. 275–305. John Wiley, Chichester.

Klein, H. and Lyytinen, K. (1985). The poverty of scientism in information systems. In Mumford, E., Hirschheim, R., Fitzgerald, G., and Wood-Harper, A. T. (eds.) *Research methods in information systems*, pp. 131–62. North-Holland, Amsterdam.

Kling, R. (1977). The organizational context of user-centred software design. *MIS Quarterly* **1** (4), 41–52.

Kling, R. (1978). Value-conflicts and social choice in electronic funds transfer developments. *Communications of the ACM* **21** (8), 642–57.

Kling, R. (1980). Social analyses of computing: theoretical perspectives in recent empirical research. *ACM Computing Surveys* **12** (1), 61–110.

Kling, R. (1985). Computerization as an ongoing social and political process. *Proceedings of Conference on Development and Use of Computer-Based Systems Tools*, Aarhus, 19–23 August, pp. 309–28.

Kling, R. and Iacono, S. (1984). The control of information systems development after implementation. *Communications of the ACM* **27** (12), 1218–26.

Kling, R. and Scacchi, W. (1982). The social web of computing: computer technology as social organization. *Advances in Computers* **21**, 2–90.

Kumar, K. and Welke, R. (1984). Implementation failure and system developer values. In Maggi, L., King, J. L., and Kraemer, K. L. (eds.) *Proceedings of the Fifth ICIS*, Tucson, Arizona, 28–30 November, pp. 1–17.

Land, F. and Hirschheim, R. (1983). Participative systems design: rationale, tools, and techniques. *Journal of Applied Systems Analysis* **10**, 91–107.

Land, F., Mumford, E., and Hawgood, J. (1980). Training the system analyst of the 1980s: four analytical procedures to assist the design process. In Lucas, H., Land, F., Lincoln, T., and Supper, K. (eds.) *The information systems environment*, pp. 239–56. North-Holland, Amsterdam.

Lanzara, G. F. (1983). The design process: frames, metaphors and games. In Briefs, U., Ciborra, C., and Schneider, L. (eds.) *Systems design for with and by the users*, pp. 29–40. North-Holland, Amsterdam.

Lanzara, G. F. and Mathiassen, L. (1985). Mapping situations within a systems development project. *Information & Management* **8** (1), 3–20.

Leavitt, H. and Whisler, T. (1958). Management in the 1980's. *Harvard Business Review*, November-December.

Legge, K. (1984). *Evaluating planned organizational change*. Academic Press, London.

Lehman, M. M. (1980). Programs, lifecycles, and laws of software evolution. *Proceedings of the IEEE* **68**, 1060–76.

Lehman, M. M. (1984). Program evolution. *Information Processing & Management* **20** (1–2), 19–36.

Lehtinen, E. and Lyytinen, K. (1986). Action based model of information systems. *Information Systems* 13 (4), 299–317.

Lientz, B. P. and Swanson, E. B. (1980). *Software maintenance management*. Addison-Wesley, Reading, Mass.

Lucas, H. C. (1973). User reactions and management of information services. *Management Informatics* 2 (4).

Lucas, H. C. (1974). Systems quality, user reactions, and the use of information systems. *Management Informatics* 3 (4).

Lucas, H. C. (1975). *Why information systems fail*. Columbia University Press, New York.

Lucas, H. C. (1978). Empirical evidence for a prescriptive model of implementation. *MIS Quarterly* 2 (1), 27–41.

Lucas, H. C. (1981). *Implementation—the key to information systems success*. Columbia University Press, New York.

Lucas, H. C. and Plimpton, P. (1972). Technological consulting in a grass roots, action oriented organization. *Sloan Management Review* 14 (1), 17–36.

Lundeberg, M., Goldkuhl, G., and Nilsson, A. (1981). *Information systems development—a systematic approach*. Prentice-Hall, Englewood Cliffs, NJ.

Lyytinen, K. (1986). Information systems development as social action: framework and critical implications. Ph.D. Thesis, Department of Computer Science, University of Jyvaskyla, Finland.

Lyytinen, K. (1987a). Two views of information modeling. *Information & Management* 11.

Lyytinen, K. (1987b). A taxonomic perspective of information systems development: theoretical constructs and recommendations. In Boland, R. J. and Hirschheim, R. A. (eds.) *Critical issues in information systems research*, pp. 3–41. John Wiley, Chichester.

Lyytinen, K. and Lehtinen, E. (1987). Seven mortal sins of systems work. In Docherty, P., Fuchs-Killowski, K., and Mathiassen, L. (eds.) *Information systems for human productivity—participation and beyond*, North-Holland, Amsterdam.

Lyytinen, K., Klein, H., and Hirschheim, R. (1986). The effectiveness of office information systems: a social action perspective. Oxford Institute of Information Management Report RDP 86/14. Templeton College, Oxford.

MacIntosh, N. and Daft, R. (1978). User department technology and information design. *Information & Management* 1 (3), 123–31.

Maish, A. (1979). A user's behavior toward his MIS. *MIS Quarterly* 3 (1), March.

March, J. and Olsen, J. (1976). *Ambiguity and choice in organizations*. Universitetsforlaget, Oslo.

Markus, M. L. (1983). Power, politics and MIS implementation. *Communications of the ACM* 26 (6), 430–44.

Markus, M. L. (1984). *Systems in organizations: bugs and features*. Pitman, Boston.

Markus, M. L. and Robey, D. (1983). The organizational validity of management information systems. *Human Relations* 36 203–26.

Markus, M. L. and Robey, D. (1986). Information technology and organizational change—conceptions of causality in theory and research. UCLA, Department of Computers and Information Systems, unpublished working paper.

Martin, J. (1985). *The information system manifesto*. Prentice-Hall, Englewood Cliffs, NJ.

Mason, R. O. and Mitroff, I. I. (1973). A program for research on management information systems. *Management Science* 19 (5), 475–87.

Mason, R. O. and Mitroff, I. I. (1981). *Challenging strategic planning assumptions*. John Wiley, New York.

Mason, R. O. and Swanson, E. B. (1981). *Measurement for management decision*. Addison-Wesley, Reading, Mass.

Mayntz, R. (1984). Information systems and organizational effectiveness—a sociological perspective. In Bemelmans, T. (ed.) *Beyond productivity: information systems development for organizational effectiveness*, pp. 173–88. North-Holland, Amsterdam.

McCracken, D. and Jackson, M. (1981). A minority dissenting opinion. In Cotterman, W. W., Couger, J. D., Enger, N. L., and Harold, F. (eds.) *Systems analysis and design: a foundation for the 1980's*, pp. 551–3. North-Holland, New York.

McFarlan, W. (1981). Portofolio approach to information systems. *Harvard Business Review* 59 (5), 142–50.

McFarlan, W. (1984). Information technology changes the way you compete. *Harvard Business Review*, May–June, 98–103.

McFarlan, W. and McKenney, J. (1983). The information archipelogo: governing the world. *Harvard Business Review*, July–August, 91–5.

McLean, E. (1979). End users as application developers. *MIS Quarterly*, 3.

Miller, G. (1956). The magical number seven—plus or minus two: some limits on our capacity for processing information. *The Psychological Review* 63 (2).

Mintzberg, H. (1973). *The nature of managerial work*. Harper & Row, New York.

Mintzberg, H. (1975). The manager's job: folklore and fact. *Harvard Business Review*, July–August, 49–61.

Mintzberg, H. (1976). Planning on the left side and managing on the right. *Harvard Business Review*, July–August.

Mitroff, I. (1980). Towards a logic and methodology for 'real world' problems. In Bjorn-Andersen, N. (ed.) *The human side of information processing*, pp. 187–95. North-Holland, Amsterdam.

Mitroff, I. and Mason, R. O. (1983). Can we design systems for managing messes? or why so many management information systems are so uninformative. *Accounting, Organizations and Society* 8 (2/3), 195–203.

Morgan, H. and Soden, J. (1973). Understanding MIS failures. *Data Base* 5 (1), Winter.

Moto-Oka, T. (1981). Challenge for knowledge information processing systems. In *Proceedings of International Conference on Fifth Generation Computer Systems, October 19–22*, pp. 1–86. Japan Information Processing Centre.

Mowshowitz, A. (1976). *Information processing in human affairs*. Addison-Wesley, Reading, Mass.

Mumford, E. (1983). *Designing human systems—the ETHICS method.* Manchester Business School, Manchester.

Mumford, E. and Pettigrew, A. (1975). *Implementing strategic decisions*. Longmans, London.

Naumann, J., Davis, G. and McKeen, J. (1980). Determining information requirements: a contingency method for the selection of a requirements assurance strategy. *Journal of Systems and Software* 1 (1).

Naumann, J. and Jenkins, M. (1982). Prototyping: the new paradigm for systems development. *MIS Quarterly* 6 (3), 29–73.

Newman, M. and Rosenberg, D. (1985). Systems analysts and the politics of organizational control. *Omega* 13 (5), 393–406.

Nunamaker, J. F. and Konsynski, B. (1981). Formal and automated techniques of systems analysis and design. In Cotterman, W. W., Couger, J. D., Enger, N. L., and Harold, F. (eds.) *Systems analysis and design: a foundation for the 1980's*, pp. 291–320. North-Holland, Amsterdam.

Nygaard, K. (1986). Program development as a social activity. In Kugler, H. J. (ed.) *Information Processing 86, Proceedings of the IFIP World Congress, Dublin*, pp. 189–98. North-Holland, Amsterdam.

Nygaard, K. and Handlykken, P. (1981). The systems development process, its setting, some problems and needs for methods. In *Software engineering environment*, Hunke, H. (ed.), pp. 157–72. North-Holland, Amsterdam.

Olerup, A. (1980). On a contingency framework of computerized information systems. In *Lucas, H., Land, F., Lincoln, T., and Supper, K. (eds.) The information systems environment*, pp. 83–94. North-Holland, Amsterdam.

Olle, T. W., Sol, H., and Tully, C. (eds.) (1983). *Information systems design methodologies: a feature analysis*. North-Holland, Amsterdam.

Olle, T. W., Sol, H., and Verrijn-Stuart, A. (eds.) (1982). *Information systems design methodologies: a comparative review*. North-Holland, Amsterdam.

Olle, T. W., Sol, H., and Verrijn-Stuart, A. (eds.) (1986). *Information systems design methodologies: improving the practice*. North-Holland, Amsterdam.

Olson, M. and Chervany, N. (1980). The relationship between organizational characteristics and the structure of the information services function. *MIS Quarterly* 4 (3), 57–68.

Palmer, I. and Gradwell, D. (1976). The key to effective use of data. *Data Processing*, July–August.

Pearson, S. (1977). Measurement of computer user satisfaction. Ph.D. Thesis, Department of Decision Sciences, Arizona State University.

Pettigrew, A. (1973). The politics of organizational decision-making. Tavistock, London.

Podger, D. (1979). High level languages—a basis for participative design. In Szyperski, N. and Grochla, E. (eds.) *Design and implementation of computer-based information systems*, pp. 243–59. Sijthoff & Noordhoof, Alphen aan den Rijn, Netherlands.

Powers, R. and Dickson, G. (1973). MIS project management: myths, opinions, and reality. *California Management Review* 15 (3), 147–56.

Robey, D. (1979). User attitudes and management information system use. *Academy of Management Journal* 22 (3), 527–38.

Robey, D. (1981). Computer information systems and organization structure. *Communications of the ACM* 24 (10), 679–87.

Robey, D. (1984). Conflict models for implementation research. In Schultz, R. L. and Ginzberg, M. (eds.) *Management science implementation*. Elsevier, New York.

Robey, D. and Markus, M. L. (1984). Rituals in systems design. *MIS Quarterly* 8 (1), 5–15.

Robey, D. and Zeller, R. (1978). Factors affecting the success and failure of an information system for product quality. *Interfaces* 8 (2).

Rockart, J. (1979). Chief executives define their own data needs. *Harvard Business Review*, 57, March–April, 81–92.

Rosenberg, R. L. (1980). Incomprehensible computer systems: knowledge without wisdom. Masters Thesis, MIT, Laboratory of Computer Science, Cambridge, Mass.

Ross, D. T. and Schoman, K. (1977). Structural analysis for requirements definition. *IEEE Transactions on Software Engineering* 3 (1), 6–15.

Rubenstein, A., Radnor, M., Baker, N., Heiman, D., and McGolly, J. (1967). Some organizational factors related to the effectiveness of management science groups in industry. *Management Science* 13 (8).

Sackman, H. (1971). Mass information utilities and social excellence. Auerbach, New York.

Salaway, G. (1984). An organizational learning approach to generating information for information systems development. Ph.D. Thesis, Graduate School of Management, Department of Computers and Information Systems, UCLA.

Sandberg, A. (1985). Socio-technical design, trade union strategies and action research. In Mumford, E., Hirschheim, R., Fitzgerald, G., and Wood-Harper, A. T. (eds.) *Research methods in information systems*, pp. 79–92. North-Holland, Amsterdam.

Sanders, G. L. (1984). MIS/DSS success measure. *Systems, Objectives, Solutions* 4, 29–34.

Saunders, D. (1974). *Computers and management*, 2nd edn. McGraw-Hill, New York.
Scacchi, W. (1985). Applying social analysis of computing to system design. In *Proceedings of Conference on Development and Use of Computer-based Systems Tools, Aarhus, 19–23 August*, pp. 477–99.
Schewe, C. (1976). The management information systems user: an exploratory behavioral analysis. *Academy of Management Journal* **19** (3), 577–90.
Schewe, C., Wiek, J., and Dann, R. (1977). Marketing the MIS. *Information & Management* **1** (1).
Schmitt, J. and Kozar, K. (1978). Management's role in information system development failures: a case study. *MIS Quarterly* **2** (2), June.
Schonberger, R. (1980). MIS design: a contingency approach. *MIS Quarterly* **4** (1), 13–20.
Schwartz, R., Fortune, J., and Horwich, J. (1980). AMANDA: a computerized document management system. *MIS Quarterly* **4** (3).
SEN (1983). Letter from the editor (1983): computers and human safety. *Software Engineering Notes* **8** (5), 7–9.
Senko, M. (1975). Information systems, records, relations, sets, entities, and things. *Information Systems*, **1** (1), 3–13.
Senn, J. (1978). A management view of systems analysts: failures and shortcomings. *MIS Quarterly* **2** (3), 25–32.
Senn, J. (1979). Enhancing the user interface: a research problem in computer based information systems. WP 75-01. School of Management, State University of New York at Binghamton.
Shackel, B. (ed.) (1985). *Human–computer interaction: INTERACT-84*. North-Holland, Amsterdam.
Shneiderman, B. (1981). *Software psychology*. Winthrop, New York.
Sibley, E. (1986). The evolution of approaches to information systems design methodologies. In Olle, T. W., Sol, H., and Verrijn-Stuart, A. (eds.) *Information systems design methodologies: improving the practice*, pp. 1–17. North-Holland, Amsterdam.
Simila, J. and Nuutinen, R. (1983). On the analysis of the user role in the context of ADP systems implementation: theoretical, methodological, and operational aspects and the first results of a case study. In *Proceedings of the 4th International Conference on Information Systems*, Houston, 15–17 December, pp. 197–221.
Sprague, R. H. (ed.) (1980). *Decision support systems. Data Base* **12** (1/2), Fall.
Srinivasan, A. (1985). Alternative measures of systems effectiveness: associations and implications. *MIS Quarterly* **9** (3).
Stamper, R. (1985). Management epistemology: garbage in, garbage out (and what about deontology and axiology). In Methlie, L. B. and Sprague, R. H. (eds.) *Knowledge representation for decision support systems*, pp. 55–77. North-Holland, Amsterdam.
Steinmuller, W. (1984). Consequences of information technology. *Information Age* **6** (3), 163–77.
Stewart, R. (1971). *How computers affect management*. Macmillan, London.
Strassmann, P. A. (1985). *The information payoff: the transformation of work in the electronic age*. The Free Press, New York.
Sullivan, C. H. (1985). Systems planning in the information age. *Sloan Management Review* **27** (1), 3–11.
Swanson, E. B. (1974). Management information systems: appreciation and involvement. *Management Science* **21** (2), 178–88.
Swanson, E. B. (1984). Organizational designs for software maintenance. In Maggi, L., King, J. L., and Kraemer, K. L. (eds.) *Proceedings of the 5th International Conference on Information Systems, Tucson, 28–30 November*, pp. 73–82.
Taggart, W. M. and Robey, D. (1980). Minds and managers: on the dual nature of human information processing and management. Florida International University, Division of Management, Working Paper.
Taggart, W. M. and Tharp, M. O. (1977). A survey of information requirements analysis techniques. *ACM Computing Surveys* **9** (1), 273–90.
Teichroew, D., Macasovic, R., Hershey, E. A., and Yamamoto, Y. (1980). Application of the entity-relationship approach to information processing systems modelling. In Chen, P. P. (ed.) *Entity-relationship approach to systems analysis and design*, pp. 15–36. North-Holland, Amsterdam.
Thompson, P. (1982). *Quality circles*. AMACOM, New York.
Turner, J. A. (1982). Observations on the use of behavioral models in information systems research and practice. *Information & Management* **5** (3), 207–13.
Ulrich, W. (1983). *Critical heuristics of social planning: a new approach to practical philosophy*. Haupt, Bern.
Vitalari, N. P. (1985). Knowledge as a basis for expertise in systems analysis: an empirical study. *MIS Quarterly* **9** (3), 221–41.
Vitalari, N. P. and Dickson, G. (1983). Problem solving for effective systems analysis: an experimental exploration. *Communications of the ACM*, **26** (11), 945–55.
Walton, R. and Vittori, W. (1983). New information technology: an organizational problem or opportunity? *Office: Technology & People* **1**.
Waters, S. (1974). Computer-aided methodology of computer systems design. *The Computer Journal* **17** (3), 211–15.
Weiss, C. (1972). *Evaluation research*. Prentice-Hall, Englewood Cliffs, NJ.
Weiss, M. (1983). The human side of systems: an experimental approach. *Information & Management* **6**, 103–8.

Westerlund, G. and Sjostrand, S. (1978). *Organizational myths*. Harper & Row, New York.
White, K. (1984). MIS project teams: an investigation of cognitive style implications. *MIS Quarterly* **8** (2), 95–101.
Williams, E. (1969). Changing systems and behavior. *Business Horizons*, July/August.
Wynn, E. H. (1979). Office conversation as an information medium. Ph.D. Thesis, Department of Anthropology, University of California, Berkeley.
Yeh, R. T., Zave, A. P., and Cole, G. E. (1980). Software requirements: a report of the state of the art. TR-949 (October). University of Texas, Austin, Department of Computer Science.
Yourdon, E. (1982). *Managing the system life cycle*. Yourdon Press, New York.
Zmud, R. (1979). Individual difference and MIS success: a review of the empirical literature. *Management Science* **25** (10), 966–79.
Zmud, R. and Cox, J. (1979). The implementation process: a change approach. *MIS Quarterly* **3** (2).