2000

Same Words, Different Meanings: Are Basic IS/IT Concepts Our Self-Imposed Tower of Babel?

Steven Alter

Communications of the Association for Information Systems

Volume 3, Article 10
April 2000

SAME WORDS, DIFFERENT MEANINGS: ARE BASIC IS/IT CONCEPTS OUR SELF-IMPOSED TOWER OF BABEL?

Steven Alter
School of Business Administration
University of San Francisco
alter@usfca.edu

FUNDAMENTALS OF IS

SAME WORDS, DIFFERENT MEANINGS: ARE BASIC IS/IT CONCEPTS OUR SELF-IMPOSED TOWER OF BABEL?

Steven Alter
School of Business Administration
University of San Francisco
alter@usfca.edu

ABSTRACT

This article began as a response to an exchange of letters concerning the need for more vs. less user participation in IS projects. It grew into an exploration of whether and how ten 1999 *CAIS* articles use basic IS/IT terms with different meanings and connotations related to the different perspectives of their authors. The article characterizes differences between an IT perspective and a business perspective and categorizes the ten articles accordingly. It then presents numerous quotes from the articles to illustrate differences across the articles in terms of their use of eight basic concepts: system, user, stakeholder, IS project, implementation, reengineering, requirements, and solution. To help understand the differences and their significance, the article makes extensive use of distinctions between work systems, information systems, and projects. When applied to the articles these distinctions raise questions such as whether the term "system" refers to a work system, information system, or software, and whether the term "user" refers to hands-on users, people who receive information, or managers whose organizations use information systems. An underlying theme throughout is that the lack of conscious attention to the meaning of basic terms and points of reference may be a significant impediment to effective communication and to our ability to make sense out of research findings and even journalistic anecdotes about what seemed to work or not work in particular situations.

Keywords: system, information system, work system, system project, project management, user, information system success

I. INTRODUCTION

I have long been convinced that a lack of clarity about basic concepts is one of the most fundamental problems in the IS field. It is difficult to define or even discuss IS principles if the meaning of underlying concepts such as system, system development, and user is so unclear that people in the same discussion might unconsciously attach different meanings to the same terms. I believe that the lack of clarity in basic concepts is a major obstacle to collaboration and communication between IS professionals, business professionals, and IS researchers.

Although this paper grew into something larger (yet another example of the "scope creep" that often happens in the IS field), I started writing it as a response to a November 1999 exchange of letters between Jim Sutter and Lorne Olfman in *CAIS* in entitled "The Case of/for the Missing User." [Sutter and Olfman, 1999]. Based on his experience as CIO of a Fortune 50 corporation, Sutter's letter expresses an idea that sounds politically incorrect: Perhaps users should NOT be so involved in software development, especially when the issues involve topics about which users have little insight or knowledge, technical issues such as Windows 2000 migration, COM vs. CORBA, selection of web servers, and so on. He argues "IT projects suffer many times from having too many cooks in the kitchen." He compares the high success rate of Y2K remediation projects with the high rate of failure or disappointment with projects that had much more user involvement, especially when user sessions "are made complex and unnecessarily contentious by debates over platforms, vendors, products, and even in release levels" in the interests of "gaining ownership and functional user 'buy-in'."

When I first glanced at Sutter's letter my immediate response was disbelief since "anyone knows that user involvement is important and beneficial." Then I took another look and realized that Sutter's users were functional area managers and their representatives, people with enough clout to become involved in discussions of technical IT strategy whether or not they had much knowledge to contribute. These are people CIOs and high level IT managers view as "their users" but these aren't the people I usually think of as users, namely, people who use information systems directly.

My first glance at Olfman's response also raised questions. He starts "Is there too much user participation in IS projects? It is possible, but without rigorous research in this domain we are just guessing." He notes that Ives and Olson [1984] concluded that research had not been able to demonstrate the value of (what they termed) end user involvement in systems development projects." Later he says, "Sutter's questioning of user participation brings to light the need for researchers to focus on finding what levels of participation are most effective. … Providing meaningful empirical studies that can help practitioners learn the key parameters for systems development success will be a welcome addition to the IS discipline's knowledge base."

My gut reaction, especially after re-reading Sutter's letter, was that adding to the IS discipline's knowledge base is NOT the issue in this particular case and that more research would NOT provide useful answers to Olfman's very general opening question, "Is there too much user participation in IS projects?" The real issue is that this question is too broad. The users under consideration might have data entry roles, information usage roles, or management roles. Their form of involvement might be symbolic involvement, involvement by advice, involvement by weak control, involvement by doing, or involvement by strong control. [Ives and Olson, 1984] Usage of the information system might be voluntary or mandatory. The IS project might involve modifying technical infrastructure, fixing technical bugs such as the Y2K bug, building an information system from scratch, installing packaged software developed elsewhere, developing a prototype, developing a small information system through an end user computing approach, hooking into an industry supply chain, and so on. There are so many different types of users, different types of involvement, different usage situations, and different types of projects that broad generalizations about whether the entire world has too much user participation in IS projects are not useful.

Upon re-reading the Sutter/Olfman letters carefully I have come to agree with a lot of very sensible things they say. At the same time, the intended controversy about excessive user participation seems artificial. Sutter's claim about too many cooks in the kitchen is about certain types of situations and is not about user participation in general. Similarly, his example of Y2K remediation is representative of technical IT projects in which users have little to contribute, but it certainly isn't representative of most IS projects. Although the two letters raised many valid issues about effective user participation, the intended controversy is less a controversy than an easily resolved confusion about vocabulary and basic concepts. Sutter was clear enough about the situations he was referring to, but my personal associations and assumptions regarding IS/IT vocabulary and basic concepts initially misled me. In contrast with Sutter's experience as a CIO, when I see the terms "user participation" and "IS project" together I typically do not think of business executives voicing uninformed opinions about inherently technical topics.

Ironically, this same type of issue about the need for a nuanced view rather than a broadbrush, one-size-fits-all, view in basic IS concepts was a major topic in another *CAIS* paper appearing same month as the Sutter/Olfman letters. This paper, "Dimensions of Information Systems Effectiveness," [Seddon et al, 1999] built upon an earlier paper [DeLone and McLean, 1992] that attempted to summarize previous research about IS effectiveness. The new paper presents a framework for evaluating IS effectiveness based two variables, the stakeholder's point of view and the type of system. It identifies five points of view for evaluating

IS effectiveness (an uninvolved observer, an individual who wants to be better off, a group which wants to be better off, managers or owners, and a country or society that wants to be better off). It then identifies six different views of what system is being evaluated (an aspect of IT use, a single IT application, a type of IT application, all IT applications used by an organization, an aspect of a system development methodology, and the IT function of an organization). The paper combines these two dimensions into a 5 X 6 matrix and argues that different measures of IS effectiveness are needed for each of the different combinations of stakeholder and system. The similarity with the relationship between user participation and IS projects is that one size does not fit all situations, whether or not the same terms, “user participation” and “IS projects,” happen to be used.

In November 1999, with around 30 years of IS research under our collective belts, these two *CAIS* articles raised issues about the definitions of basic IS concepts including user, user participation, IS project, and system. I think this is a cause for concern about what Olfman’s last sentence refers to as “the IS discipline’s knowledge base.” At minimum, I think the range of different meanings and connotations associated with the most commonplace IS terms implies that we as a field need to pay more attention to basic concepts and how different people use them.

Instead of talking in the abstract about the need for clearer concepts or greater rigor, this article uses ten other *CAIS* articles to explore whether basic concepts are actually used differently and whether these differences might detract from the effectiveness of “the IS discipline’s knowledge base.” This article’s unifying principles are based on “A General, Yet Useful Theory of Information Systems,” [Alter, 1999a]. Since that article is lengthy, the pertinent ideas are summarized in Appendix I. These ideas include:

- the definition of work system as a system in which human participants and/or machines perform a business process using information, technology, and other resources to produce products and/or services for internal or external customers

- the fact that information systems and projects are special cases of work systems
- the phases of a project (initiation, development, implementation, maintenance)
- the differences between work system projects, information system projects, and software projects
- the fact that information systems and projects should both "inherit" generalizations about work systems because they are special cases of work systems.

The first major section uses these ideas to characterize the way basic concepts might be seen from two different viewpoints which are called the IT perspective and the business perspective. The next section explores whether the difference between these viewpoints has much bearing on the way authors use basic IS/IT concepts. It identifies ten articles published between June and December 1999 in Volume 2 of CAIS. Four articles seem closer to an IT perspective and six seem closer to a business perspective. Six emphasize system projects and four emphasize systems in operation. A comparison of representative statements from the articles shows that basic IS/IT concepts such as system, project, requirements, user, and reengineering are used with different meanings and connotations.

The outline of the paper is as follows:

II. IT PERSPECTIVE VS. BUSINESS PERSPECTIVE

Whether IS researchers and practitioners focus on software or IS plumbing, on the one hand, versus IS content or work systems on the other, often boils down to whether they are taking an IT perspective or a business perspective. Each perspective is valid. The confusion ensues when it isn't clear which perspective is being taken or when one perspective is assumed to be obvious and the other is ignored even though it might be worthwhile to consider. (Later, this paper will look for relevant examples in other *CAIS* papers.)

Table 1 describes the basic viewpoint of people who take an IT perspective versus a business perspective. The characterizations in the Table are stereotypes that highlight or possibly exaggerate likely biases in an individual's viewpoint. Obviously, highly conscious practitioners in each realm appreciate the need to collaborate with practitioners from the other realm and often learn enough about the other realm to be able to communicate effectively in that other realm's own terms. Less conscious practitioners in each realm tend stick closer to their own knitting and sometimes stereotype the other realm along the lines that are presented in the Table. (See [Beath and Orlikowski, 1994] for an exploration of some of these issues using the literary technique of deconstruction to analyze contradictions related to users in system development methodologies.)

Whereas Table 1 tried to characterize the people who tend to have an IT perspective or business perspective, Table 2 summarizes the primary differences between the perspectives in terms of meanings and connections applied to eight basic IS concepts. From an IT perspective, "the system" is primarily software or the computerized parts of an information system. The system's goal is to operate consistent with unambiguous specifications and within budget. The

Table 1: Characterizations related to core viewpoints for people who typically take an IT perspective vs. business perspective.

Aspect of core viewpoint	*Typical characterization of people with an IT perspective*	*Typical characterization of people with a business perspective*
What is my professional affiliation?	IT or IS practitioner or computer scientist	Non-IT profession such as general management, sales, finance, production, engineering, law, medicine, etc.
What do I really care about in relation to information systems?	Creating and maintaining information systems that meet requirements and operate within budget.	The role of information systems within work systems that accomplish their goals, that provide a rewarding work life for me, and allow us to compete effectively
Negative stereotypes about people in the opposite realm	People in the business realm often know very little about my technical realm and have unrealistic expectations about what is possible or practical. They always want an instant answer.	People in the IT realm tend to be techies who focus on their own technical work and don't care very much about whether I succeed at my work.
What should I try to remember when collaborating with people from the opposite realm?	Try to stay cognizant of the work systems and political issues that business colleagues must deal with every day.	Try to stay cognizant of the limits of IT, especially the limited ability to change IS plumbing rapidly.
Attraction of technical work	Enjoy doing technical work, or at minimum is highly competent doing technical work	May enjoy doing technical work, may tolerate, or may hate it.

users are the people who use computerized devices directly, the people who use information generated by information systems, and organization managers. From a business perspective "the system" can mean any of the following, and sometimes takes on all four meanings in the same conversation:

- It may be a computer, as in "I went to Office Depot and bought a Compaq system."
- It may be software, such as Oracle's Finance module.
- It may be an operational information system, such as XYZ Company's system for closing the books.
- And it may be a system that involves data processing and many other things, such as XYZ Company's system for delivering automobiles.

With an IT perspective the main concern is the work of IS professionals. This perspective tends to view IS projects as the work of contractors operating under fixed price contracts and therefore very concerned about obtaining precise requirements and holding fast to those requirements, or at minimum, identifying

Table 2: IT perspective vs. business perspective on basic IS concepts

Topic	*IT perspective*	*Business perspective*
System	The system is software or the parts of an information system that are computerized.	The system is a work system in which human participants and/or machines perform a business process using information, technology, and other resources to produce products and/or services for internal or external customers.
User	Someone who uses computer software or information produced or transmitted using IT.	Someone who uses an information system (either the software or the information that is generated) as part of being a participant a work system whose practices dictate whether information system usage is mandatory or voluntary. Being a successful work system participant is typically a far more important concern than information system features or capabilities.
Stakeholder	Same for both perspectives: Someone who is affected by an information system and whose organizational role and/or status permit direct or indirect involvement in the determination of how systems should operate.	Same for both perspectives: Someone who is affected by an information system and whose organizational role and/or status permit direct or indirect involvement in the determination of how systems should operate.
IS project	A project whose immediate goal is building or modifying software. Declare victory when the software meets requirements and is accepted by users or their managers.	A project whose immediate goal is improving a work system. Declare victory when the work system meets operational goals and has a mechanism for continuing successful change
Implementation	Transforming requirements into software that operates correctly on the computer and therefore satisfies the requirements.	Making a new or modified business process fully operational in an organization.
Reengineering	Revamping the technical components of an information system.	Making substantial changes in business processes in order to make the business processes more efficient and/or effective.
Requirements	Unambiguous statement of the processing that should be done by an information system in order to provide the agreed upon benefits to the users. Clear requirements are needed as part of system development before programming begins.	Excessively detailed statements about desired processing, unrealistically cast in concrete so that it will be possible to evaluate whether programmers completed their work on time and within budget. IT people sometimes use requirements as an excuse for not fixing inadequate programs completely.
Solution	Purchased software and/or hardware capable of performing particular data processing functions.	A way to solve a business problem. Unless the business problem is about inadequate or malfunctioning technology, a solution almost always includes major features and capabilities beyond those built into software and/or hardware.
Point of reference (for someone using each perspective)	Take the viewpoint of an outside contractor doing a project for someone else. Declare victory when the information system operates consistent with requirements and is maintainable.	Take the viewpoint of a business manager or of a business professional who is a participant in a work system. Declare victory when the work system meets operational goals on an on-going basis and provides an appropriate work environment and personal growth for its participants.

requirements creep and enforcing agreements about how to change the contract when the requirements inevitably change. In contrast, a business perspective emphasizes the view of business professionals who see information systems as a necessary part of the current and future operation of work systems they participate in or manage. Depending on their management level they may or may not be concerned with project budgets, but many of them are concerned about the difficulty in operating information systems as beneficial tools rather than as obstacles that make it difficult to do work in the most effective way. They recognize themselves as information system users, but their various roles as work system participants or managers is more important to them than their role as information system users.

III. IT VS. BUSINESS PERSPECTIVE AND THE USE OF BASIC CONCEPTS: EVIDENCE FROM *CAIS* ARTICLES

This section looks for evidence of the IT perspective and business perspective in a selected set of ten articles in Volume 2 of *CAIS* published between June1999 through December 1999. The ten articles were selected because each covers topics that might be approached from either perspective or from a combination of perspectives. They tend not to include articles whose main topic is a single case study, a research method, a particular technology, or the state of IS in academia. By coincidence, only one of these papers ([2(17)] below) was written in the United States. Seven of them papers were edited by Chris Holland and are part of the Esprit project in the United Kingdom. The other two are from Greece [2 (15)] and Australia [2(20)]. Whether or not nationalities of the authors affect the results of this analysis is a matter of conjecture that might be tested by performing this type analysis using a different group of articles.

Table 3 summarizes the topic of each paper and categorizes it in terms of perspective and emphasis on system projects or system operation. It shows that four articles seem closer to an IT perspective and six seem closer to a business

Table 3: Main Topics of Ten Articles in Volume 2 of CAIS

System project	• 2(03) Migrating to object/ component technology • 2(17) Software project management • 2(24) Using patterns for reengineering	• 2(04) Recognizing importance of stakeholders • 2(05) Linking business process design and IS design • 2(06) Modeling organizational change using the EKD framework
Systems in operation	• 2(15) Evaluating a document repository	• 2(07) Business aspects of legacy systems • 2(08) Observing legacy technology in use in a bank • 2(20) Evaluating IS success
	IT perspective	*Business perspective*

2(03) O'Callaghan, A. J. "Migrating Large-Scale Legacy Systems to Component-Based and Object Technology: The Evolution of a Pattern Language," *Communications of AIS*, Vol. 2, Article 3, July 1999.

2(04) Coakes, E. and T. Elliman. "The Role of Stakeholders in Managing Change," *Communications of AIS*, Vol. 2, Article 4, July 1999.

2(05) Giaglis, G. M. "On the Integrated Design and Evaluation of Business Processes and Information Systems," *Communications of AIS*, Vol. 2, Article 5, July 1999.

2(06) Kavakali, V. and P. Loucopoulos. "Modelling of Organisational Change Using the EKD Framework," *Communications of AIS*, Vol. 2, Article 6, July 1999.

2(07) Kelly, S., N. Gibson, C.P. Holland, and B. Light. "A Business Perspective on Legacy Information Systems," *Communications of AIS*, Vol. 2, Article 7, July 1999

2(08) Randall, D., J. Hughes, J. O'Brien, T. Rodden, M. Rouncefield, I. Sommerville, and P. Tolmie. "Banking on the Old Technology: Understanding the Organizational Context of 'Legacy' Issues," *Communications of AIS*, Vol. 2, Article 8, June 1999.

2(15) Metaxiotis, K.S., A. P. Papakonstantinou, J.E. Psarras. (1999): "Evaluating the Integrated Measurement and Evaluation System IMES: A Success Story," *Communications of AIS*, Vol 2, Article 15, September 1999.

2(17) Jurison, J. (1999) "Software project Management: A Manager's View, " *Communications of AIS*, Vol. 2, Article 17, September 1999.

2(20) Seddon, P. B., S. Staples, R. Patnayakuni, and M. Bowtell. (1999) "Dimensions of Information System Success," *Communications of AIS*, Vol. 2, Article 20, November 1999

2(24) Lloyd, A.D., R. Dewar, and R. Pooley. "Legacy Information Systems and Business Process Change: A Patterns Perspective." *Communications of AIS*, Vol. 2, Article 24, December 1999.

perspective. Six emphasize system projects and four emphasize systems in operation.

Before I thought of characterizing the articles in this way (i.e., while I was skimming the articles to see how they used basic IS concepts), I tried to find at least several statements from each article that represented the way the article seemed to deal with the eight IS concepts that Table 2 interprets from extreme versions of the IT perspective and the business perspective. The concepts are: system, user, stakeholder, IS project, implementation, reengineering, requirements, and solution. I also looked for a few statements that seemed to exemplify each article's point of reference in dealing with its overall subject matter. In cases where a concept such as system or requirement was used both in a very general, colloquial sense, and in a sense specific to the article's viewpoint, I tried to choose the statements related to the article's viewpoint. In cases where a concept such as reengineering was not a major topic in the article I used the word processor's "find" command to find any usage of the term or to show that it was not used. The representative statements selected for each article are listed in Appendix II. (Obviously other readers might have selected other statements that represent any article's point of view more accurately.)

An initial comparison of representative statements from the articles suggested that the differences between the perspectives sometimes led to differences in the meanings and connotations of the basic IS concepts. For example, an article with an IT perspective might view the software as the system. In some cases it might explicitly say "software system," but in others it might just use the term system and assume that the reader knows that the system is the software.

I used the following method to explore the treatment of each of the IS concepts. First sort the statements so that all the statements about each IS concept are together. Within each concept, group the statements from the six articles with a business perspective and from the four articles with an IT perspective. Go back to the characterizations in Table 2 and see whether the statements in each group of articles seem consistent with the characterizations, which were written before the statements were analyzed.

Listed in the following are comments and examples for each of the eight IS concepts. The discussion for most concepts starts by restating the generalization in Table 2 concerning the difference between an IT perspective and a business perspective on the concept. The rest of the discussion cites some of the statements from the articles to see whether articles written from different perspectives attach different meanings and connotations to the same terms. To simplify the discussion, the articles from an IT perspective are called T articles and those with a business perspective are called B articles. Appendix II lists all of the statements selected for each article and groups those statements by topic. Appendix III lists all the statements selected for each topic and groups those statements by article.

Caveat: These characterizations and comments are a matter of opinion and are necessarily brief. Someone else's careful reading of some of the papers would certainly identify important points that are omitted, including some points that contradict the comments presented here.

SYSTEM

According to Table 2, people using a T perspective would tend to view the system under consideration as software or the parts of an information system that are computerized. In contrast, people using a B perspective would tend to view the system as a process or an organization that uses software or computerized tools. For example, since the theory of information systems summarized in Appendix I has a B perspective, someone using it would typically think of "the system" as a work system in which human participants and/or machines perform a business process using information, technology, and other resources to produce products and/or services for internal or external customers.

Overall, the usage of the term system in the B articles and T articles was more similar than polarized definitions in Table 2 might have predicted. Most of the B and T articles recognized important roles of information systems in organizations and most also viewed the information system as a separable technical resource. Although most articles recognized both sides, the B articles emphasized the process aspects of systems rather than the technical aspects:

- The role of information systems in influencing and enabling organisational design is widely acknowledged. Yet limited attention is paid to the theoretical legitimacy and conceptual basis of IS-enabled organisational change. [2(05)]
- Contemporary IS are increasingly integrated together, making it even more difficult to disentangle a single system for evaluation. This may render the demarcation of boundaries around individual systems for the purposes of evaluation a meaningless exercise. [2(05)]
- Legacy information systems are defined as information technology (e.g. hardware, software applications and network) and the business model implicit in the application of that technology (e.g. organizational structure, work flows, procedures and processes) within the organization. [2(07)]
- The real value to the organization of information technology legacy systems lies in the "accumulation of years of business rules, policies, expertise and 'knowhow' embedded in the system. [2(07)]
- …Understanding 'legacy' and its impact on business 'processes' and everyday working may require a nuanced view of various factors, including working practice, communication and control problems, and indeed any number of complex articulations of structure, process, technology, and 'situated' knowledge. [2(08)]
- 'Legacy', we argue, is not just a problem encountered by organisations with aging mainframes and dated software, it is an issue from the moment a computer system becomes an integral part of any organisation's everyday work. [2(08)]

The main topics in the T articles focused on systems as technical artifacts built by IT professionals:

- Expertise in shifting legacy systems to new paradigms is buried in the folklore of software engineering. [2(03)]
- IMES is an integrated information system that incorporates Internet technologies to provide wide monitoring and evaluation capabilities. It consists of five individual, but interacting, subsystems … the database, the local application, the input/output assistants,

the intranet component, and the security mechanism. [2(15)]

- To most users, the interface is the system. [2(15)]
- Conducted at both unit and system level, [technical reviews] are used to verify the functionality and quality of the system. [2(17)]

The following excerpts exemplify the way T articles recognized business issues even though they viewed information systems as bounded technical systems, rather than vaguely bounded systems in organizations:

- The planning cycle starts with firming up the goals and objectives and determining the requirements for the system. [2(17)]
- In many cases it may be necessary to build and test a prototype to develop a good understanding of the system's needs and requirements. [2(17)]
- The design of large IT systems is extremely hard to separate from the design of business processes. The question then arises: can legacy computer systems 'lock-in' inefficient or even redundant 'legacy' business processes? [2(24)]
- A system architect, however, lies between the Business and Technology strategists and the application programmer. Although they are responsible for designing systems to support the business strategy and will usually be aware of any constraints that the technology strategy imposes, their knowledge of the specific business strategy, and hence the factors of competition, is likely to be less detailed. [2(24)]

USER

The difference between users and other stakeholders was the basis of my misunderstanding of Sutter's letter mentioned at the outset. Table 2 defines user from an IT perspective as "someone who uses computer software or information produced or transmitted using IT." The definition from a B perspective emphasizes the fact that IS usage occurs as an aspect of participation in a work system. This implies that being a successful work system participant should be more important to users than information system features or capabilities.

Neither the B nor the T articles considered the distinction between information system usage and work system participation. Both the B and T articles seem to view users as people who use information or information systems, but except in two articles [2(04) and 2(20)] whose main points were about stakeholders, the term *user* is often applied in contexts where it is unclear whether users, other stakeholders, or entire organizations were being discussed.

- Many systems still fail to fulfill the needs of their users and the organisations that adopt them. [2(05)]
- Computer systems have been installed in many companies for some time now and no matter how well they may have fitted the situation initially, usage and the circumstances of use have changed, as indeed have the needs and the users, and, most importantly, the organisations themselves. [2(08)]
- …The development team was able to capture the essence of the business problem by working with the users. [2(03)]
- User involvement, particularly during the planning phase, leads to better and more realistic definition of system requirements and user commitment to the project. [2(17)]

The T articles tend to emphasize the interests of IT professionals and therefore show a tendency to view the users with more distance than the B articles. For example:

- …The development team was able to capture the essence of the business problem by working with the users. [2(03)]
- [Evaluation criteria include:] - Reliability: The extent to which the clients can trust the system and its services; Accessibility: The degree to which the system database is easy to be accessed by the users; Ease of use: The extent to which the users can "navigate" in the system database and use its services. [2(15)]
- Users are often unsure of their needs and frequently change requirements midway through the project. As a result, the software industry is plagued by cost overruns, late deliveries, poor reliability, and user dissatisfaction [2(17)]
- User involvement, particularly during the planning phase, leads to better and more realistic definition of system requirements and user commitment to the project [2(17)]

STAKEHOLDER

Two of the ten articles [2(04) and 2(20)] focus extensively on stakeholders.

- For our purposes a stakeholder is someone who has an interest in a CIS development *and* can affect the success of that development. [2(04)]
- [The following themes] capture a pragmatic dimension to justifying a "stakeholder's" participation.
  - Stakeholders may affect realisation or may be affected by realisation of a system;
  - Stakeholders may have actual versus legitimate influence; they may be an internal

affect or versus external;
- Stakeholders may have a supportive influence versus conflictive influence
- They may be stakeholders of a common value; they therefore need to be considered, consulted, participative, or responsible for process under consideration or development by the system. [2(04)]

- A particular risk is that close to the technical boundary we will find stakeholders who have extreme views of existing, or legacy systems. The danger is that inappropriate factors may be given more weight than the wider needs of the organisation and its environment. Stakeholders close to the technology can be expected to express their personal investment in the current technology, their detailed experience of operational problems, or their technological bias lending enthusiasm for the promises of new technology. [2(04)]

- Five different types of stakeholders might be considered when evaluating IS success: the independent observer who is not involved as a stakeholder; the individual who wants to be better off; the group, which also wants to be better off; the managers or owners who want the organization to be better off; the country which wants the society as a whole to be better off. [2(20)]

All six of the B articles mention stakeholders directly or indirectly, such as in the first three excerpts below. The mention of stakeholders in the T papers other than 2(17) is mostly indirect.

- Project champions tend to underestimate costs and overestimate benefits. [2(05)]

- The evaluation data provided organisational stakeholders with a rationale means of making an informed choice. [2(06)]

- The inclination and acceptance of change does not exist within the culture of the organization and hence employees resist change. Although managers see change as an opportunity to strengthen the business, employees may perceive change as disruptive and intrusive.[2(07)]

- A project can be considered a success only if the client, whether it is a group of internal users or a client in another company, is satisfied with the results.[2(17)]

IS PROJECT

Table 2 characterizes the difference between a B perspective and a T perspective on IS projects in terms of the overall goal. With a B perspective, the project goal is improving a work system. The goal is achieved when the work system meets operational goals and has a mechanism for continuing successful change. With a T perspective, the project goal is building or modifying software. The goal is achieved when the software meets requirements and is accepted by users or their managers. These distinct perspectives can be seen in the excerpts that follow.

The excerpts from B articles show that discussions of projects emphasized work system and organizational issues, rather than building software:

- Computer Information Systems development often focuses on direct users and affected internal departments as the exclusive stakeholders. However these groups may present too narrow a perspective. To improve the effectiveness of the development process, a wider constituency should be considered that includes organisational partners in the wider business environment. [2(04)]

- SDLC-based IS development methods …perpetuate the distinction between the business and the IS domain. Most structured approaches to IS development begin with an implicit assumption that the business domain issues are resolved and the system is to work in a stable and well-defined business environment, where the only issue is to identify the 'correct' requirements for the new IS. As a result, not enough attention is generally being paid to investigating the interactions of the IS to be developed with the business processes it will naturally affect. [2(05)]

- Change goals form the requirements upon which the re-engineered enterprise structure will be based. This task concerns the mapping of change requirements onto a future enterprise model, which in turn involves the modelling of the future enterprise goals and how these goals will be realised in terms of operational enterprise components. [2(06)]

- Legacy systems comprise the existing characteristics of an organization such as its structures, processes, strategy and cultures resulting from the impact of internal and external forces. These characteristics can mean that even when new emerging technologies are introduced into organizations, employees have difficulty in adapting to new ways of working. [2(07)]

- It is unlikely that any organisation is ever 'going to get it right' the first time. What it *does* suggest is the need for more effective monitoring of new technologies in their situations of use and developing effective mechanisms for involving users' experiences in development [2(08)].

In contrast, the T articles did focus on issues related to building software.

- The *raison d'être* for contemplating a move to an object-based representation for an existing system is the belief that business benefits in terms of increased flexibility to business change, and increased productivity (through software reuse) will result. [2(03)]

- Legacy information systems are typically the targets of reverse engineering projects. [2(03)]

- Projects have specific objectives. Projects must be completed within a specific time period. They have well defined beginnings and ends. Projects must be completed within a given budget. Although some projects may have loosely defined budgets, all projects have budgetary constraints. [2(17)]

- In information systems projects, performance is specified in terms of certain functional and quality requirements, some of which are quantitative, some qualitative. [2(17)]

- If a new system is developed to replace part of the old one, the developers will be expected to provide ideal functionality. Consequently, it will be impossible to manage expectations and the project will become large and risky. [2(24)]

IMPLEMENTATION

According to Table 2, a B perspective on implementation emphasizes making a new or modified business process fully operational in an organization, whereas a T perspective focuses on transforming requirements into software that operates correctly on the computer and therefore satisfies the requirements. The usage of "implementation" in some articles was consistent with this distinction, but there were several exceptions.

These excerpts from B articles illustrate the use of "implementation" from a B perspective:

- Another failing in the group's decision making was the difficulty in separating technology and implementation from strategic decision making. [2(04)]

- The process scenarios were scrutinised to develop a detailed understanding of implementation challenges and transform hypotheses into detailed implementation plans. The requirements of each option regarding technology, people, and skills were assessed and a formal cost-benefit analysis was conducted to evaluate the proposed investments. [2(05)]

- The implementation comprises introducing customers profiling, minimising delay time to serve an application, offering all means for payment, offering all services at customer premises, introducing all available technologies to communicate with customers, introducing IT solutions for all services. [2(06)]

- The existence of legacy systems that hinder the implementation of new business strategies is now well established. [2(07)]

- No matter how promising new technologies may seem, the realities of their implementation are typically disruptive. [2(08)]

These excerpts from T articles illustrate the use of "implementation" from a T perspective:

- When a shift is being contemplated from, say, representation in a structured language to representation in an object-oriented implementation, it is not just the language that is changing but the development paradigm itself. [2(03)]

- This scoping of the analysis model so that it captured the key abstractions of the problem space and modelled them separately and independently of any implementation concerns reflects the *Shamrock* pattern of the ADAPTOR language. [2(03)]

- The following principles are fundamental to the design and implementation of effective interfaces, either for traditional GUI environments or the Web. [2(15)]

- This pattern was used during a *Middleware* implementation. [2(24)]

The use of implementation in one of the T articles was consistent with a B perspective.

- For IS projects, the execution phase frequently extends beyond delivery of the end product and includes system implementation, the process of putting the system into operation in the client's organization. It is not uncommon to have system implementation handled by a separate project team because the implementation team often must function as a change agent rather than as a developer. [2(17)]

- Increasingly IS project managers find themselves playing a central role in their organizations, whether it is an enterprise resource planning (ERP) system implementation, Year 2000 conversion, or a leading-edge technology project. [2(17)]

In two other articles, one B and one T, the use of "implementation" either combined both perspectives or was unclear in terms of perspective.

- This point reinforces our earlier argument for incorporating only the high-level organisational impacts of IS in business process design and leaving the low-level technical implementation details for later. [2(05)]

- Finally, SDLC-based approaches tend to view IS evaluation as a post-implementation activity, addressed only in the last step of the system development life cycle. [2(05)]

- This pattern was used during a *Middleware* implementation. Note that in an organisation-specific reengineering pattern catalogue, this section would also contain contact details of managers involved in the cited implementation. [2(24)]

REENGINEERING

According to Table 2, a B perspective on reengineering would emphasis making substantial changes in business processes in order to make the business processes more efficient and/or effective. In contrast, a T perspective would emphasize revamping the technical components of an information system. Of the ten articles, four mentioned reengineering directly and one mentioned "business engineering."

Several B articles and one of the T articles [2(17)] viewed reengineering and similar topics from what seemed to be a B perspective:

- Business engineering is defined here as the integral, concurrent design of organisational processes and the information systems to support them. [2(05)]

- Any type of change whether it involves the development of a computerised system or the re-engineering of business processes involves many assumptions about the embedding enterprise domain. [2(06)]

- In significant respects, problems such as these [difficulty modifying systems already in use] are as much organisational as technological because they direct attention to the need to reorganise work and implement new technologies in a more integrated way. [2(08)]

- As an increasing number of new IS projects become more strategic and involve business process reengineering, management of organizational change is an integral part of project management. [2(17)]

Several of the T articles viewed reengineering from a T perspective.

- The migration of legacy systems is a process of re-engineering. The accepted definition of re-engineering is … "the examination and alteration of the target system to reconstitute it in a new form". [2(03)]

- There is a particular quality to the re-engineering effort that must be understood when it involves moving a computer system from, say, a structured representation to an object-based one, however. [2(03)]

- This theme of re-use is one of a number of guiding principles for developing systems reengineering patterns. [2(24)]

- (Four reengineering patterns are discussed: divide and modernise, wrapping, middleware, and externalising an internal representation. These patterns involve reengineering software rather than business processes..) [2(24)]

REQUIREMENTS

The way Table 2 looks at requirements emphasizes the battle lines between the B and T perspective. From a T perspective, requirements are an unambiguous statement of the processing that should be done by an information system to provide the agreed upon benefits to the users. Clear requirements are needed as part of system development before programming begins. From a B perspective, they are excessively detailed statements about desired processing, unrealistically cast in concrete so that so that it will be possible to evaluate whether programmers completed their work on time and within budget. Most of the B articles mentioned requirements in a non-controversial way, although one of the B articles and three of the four T articles noted some of the tensions related to requirements that are either to inflexible or too changeable.

Four of the B articles mentioned requirements in a non-controversial tone:

- [In relation to a case study] Many of these external stakeholders are governmental bodies whose needs for the supply of data and reports are an integral requirement of any university's student record keeping package. [2(04)]

- Change goals form the requirements upon which the re-engineered enterprise structure will be based. [2(06)]

- These sessions resulted in the specification of both internal enterprise needs as well as external constraints that defined the enterprise change requirements. [2(06)]

- In addition to date and regulation requirements, a range of business pressures are increasingly significant today. [2(07)]

- The two managers had to work together to produce some kind of model that seemed to give due consideration to their own, highly particular requirements. [2(08)]

Another B article noted the tension between the B perspective and the T perspective:

- Most structured approaches to IS development begin with an implicit assumption that the business domain issues are resolved and the system is to work in a stable and well-defined business environment, where the only issue is to identify the 'correct' requirements for the new IS. [2(05)]

- Although most existing IS development methods begin by stressing the importance of understanding the real-world operation that the IS will support, they quickly become absorbed in the definition of individual functions and detailed requirements ('reductionism'). [2(05)]

All four of the T articles mentioned requirements and three of them noted some of the related difficulties. A tutorial on software project management [2(17)] was especially concerned about using requirements effectively.

- A software system that tries to meet the requirements of all possible scenarios will almost certainly suffer 'analysis paralysis' and will be too complicated and/or inefficient to deliver and use. [2(03)]

- In each case the systems' owners made a business decision that they needed a component-based architecture in order to meet the challenge of ever new requirements, and this architecture implied the kind of encapsulation that object-based systems deliver. [2(03)]

- The team collected and studied the requirements of the system as defined by the main client (European Commission) and set the key evaluation questions. [2(15)]

- Users are often unsure of their needs and frequently change requirements midway through the project. As a result, the software industry is plagued by cost overruns, late deliveries, poor reliability, and user dissatisfaction. [2(17)]

- [Walkthroughs and inspections] are effective for early detection of errors in requirements, interface prototypes, design, code, and documentation. [2(17)]

- A clearly defined requirements specification, agreed upon by both the client and the development team, ensures that the client's needs are correctly understood before starting design work. The requirements document is, in effect, a contract between the client and the development team. It specifies what the product must do, but not how. [2(17)]

- Even the best prepared requirements specifications will require changes as the software is being designed and tested. Many projects fall victim to "scope creep" caused by uncontrolled changes made well beyond the requirements definition phase. [2(17)]

- Even if a requirements explosion does overtake the final restructuring step, the main aim, that of removing the dependency of the functionality on the obsolete technology, will have been achieved. [2(24)]

SOLUTION

The term "solution" was not on the original list of IS concepts. I decided to include it when I saw that it was used in several articles to denote software and/or hardware purchased from a vendor. I personally feel very uncomfortable with this usage because I believe it is misleading and often self-contradictory. IT vendors may claim they sell solutions, but unless the problem is purely in the realm of software and hardware, their solutions are at best only part of a way to address a business problem and often bring additional problems with them. For example, CEOs who have suffered through SAP implementations would probably express strong views if they could see videotapes of early presentations that presented SAP as "a solution." Regardless of my opinion about this usage of the term "solution," it seems to be creeping into the IS literature from its source in the world of marketing hype. The term "solution" was used in this sense by two B articles and three T articles:

These excerpts illustrate this use of "solution" in the two of the B articles:

- Initial planning was at a level of detail that had to be discarded when the emergence of Internet technology and applications provided a readily implementable solution in 1995. [2(04)]

- Developments in information technology add to the problem [of enhancing existing systems] as technology moves beyond traditional transaction processing towards client/server architectures and the Internet to create new types of business solutions. [2(07)]

- The company decided to reengineer the organization including implementation of a process-oriented ERP solution. [2(07)]

These excerpts illustrate this use of "solution" in the three of the T articles:

- Similar applications running on different operating systems on different boxes became common. Worse still, key business abstractions such as 'Customer' could be running on different applications on the same machine at the same time, and since these applications could not talk to each other, information integrity could not be maintained. Subsequently, such point solutions became subject to localized optimizations, and uncontrolled maintenance, etc., exacerbating the position even further. [2(03)]

- But these benefits rely, as we have seen, on the fact that object systems 'break' from the underlying Von Neuman architecture of the machine and enable the possibility of building software solutions in the image of the problem space itself. [2(03)]

- Project management packages range from simple schedulers to enterprise-wide solutions and vary in price from about $50 to several thousand dollars. [2(17)]

- The rise of pre-packaged solutions to common business processes such as accounting and invoicing, produced with economies of scale and benefiting from compliance with complex legislation, changed the equation. Smaller companies now found an economic incentive to 'fit' their business process to the standard solution. Large companies …also saw standard solutions provided by market leaders as a means of benchmarking best practice. [2(24)]

IV. DISCUSSION

The previous section presented quotes from ten1999 *CAIS* articles to illustrate the range of different meanings and connotations these articles applied to eight common IS terms: system, user, stakeholder, IS project, implementation, reengineering, requirements, and solution. Classifying the basic perspective of each article as either a "business perspective" or an "IT perspective" made it possible to show examples illustrating a tendency for articles to interpret basic IS concepts in terms of the perspective they use.

The mere fact that different articles appearing in *CAIS* in the last half of 1999 attach different meanings and connotations to the same basic concepts is both expected and disturbing. It is expected because the field is relatively immature and because information system research spans technical and behavioral disciplines and viewpoints. It is disturbing because it implies that any real attempt to accumulate a "body of knowledge" will encounter a lot of confusion, especially wherever the topic or perspective is neither purely technical nor purely behavioral.

LIMITATIONS

Before moving on to several aspects of the main topic, the interpretation and use of basic IS concepts, it is worthwhile to mention several of the most obvious limitations of the methods and results presented thus far. These limitations include the sample itself, the way the statements were selected, and the imprecise interpretation of how the statements illustrated differences in concept usage and differences between the perspectives.

The Sample

The sample was small and rather arbitrary since it focused on one journal, CAIS, and on selected articles published between June and December, 1999. As was mentioned earlier, only one of these papers was written in the United States. Seven of them papers were edited by Chris Holland and are part of the Esprit project in the United Kingdom. The other two are from Greece and Australia. Clearly it would be possible to expand the sample and make it more representative geographically by looking at articles across several years in the three or four of the leading journals. The main reason why this was not done is that this paper started as a response to a November 1999 letter in *CAIS* that led me to wonder whether some of my confusions in interpreting that letter might also occur when I looked at other articles in *CAIS* around the same time. While the sample was small and rather arbitrary, it was large enough to accomplish the purpose of illustrating inconsistencies related to basic concepts.

The Selection of Statements

The selection of statements illustrating the use of the concepts was also based on one person's reading and interpretation of the articles. Other readers might have selected other representative statements, and these might have led to different conclusions.

Imprecise Interpretation

The interpretations in the previous section are both subjective and imprecisely sketched. The little interpretation presented for most concepts boils down to "look at these quotes" and "look at those quotes" and "gee, they do look somewhat different, don't they." Given the diverse nature the topics, perspectives, and writing styles, I think that this approach was adequate for making the point that some of the articles published around the same time in the same journal use many basic concepts at least somewhat differently. A more precise statement and possibly a quantification of this effect might be feasible, but I think it is more interesting to delve into the usage of some of the basic IS concepts in these recent CAIS articles.

HOW THE THEORY HELPED IN UNDERSTANDING THE ARTICLES

One of the benefits of starting with a theory is that it highlights some topics and issues, places other topics and issues in relation to main ones, and totally ignores others. This automatically creates a framework for sorting out the topics and issues within any article and for comparing across articles.

The theory of is summarized in Appendix I uses “work system” as a basic unit of analysis. The six elements for understanding a work system at even the simplest level include the business process, participants, information, technology, product, and customer. Information systems and projects are special types of work systems. An information system is a work system whose internal functions are limited to processing information. Information systems exist to produce information and/or to support or automate the work performed by other work systems. They may serve other work systems through a variety of roles. A project is a time-limited work system designed to produce a particular product and then go out of existence. Software projects, information system projects, and work system projects are related but differ in scope and breadth of objective.

Just these observations made it easier to understand the articles and to interpret which potentially pertinent topics were and were not included. For example, the tutorial on Software Project Management [Jurison, 1999] was clearly about software projects and took a project manager's viewpoint, but also included a number of observations demonstrating an appreciation of other concerns such as implementation of the information system in the organization. In contrast, the article on evaluating an information system that was basically a document repository [Metaxiotis et al, 1999] focused strongly on evaluating the information and its potential availability, but said little about how this document repository was actually used and whether it had a significant impact on a work system.

RANGE OF MEANINGS FOR BASIC CONCEPTS

The theory of IS also made it easier to perceive how the same basic IS concepts were indeed used with different meanings in different articles.

System

Most articles viewed "the system" as an information system. Some focused mostly on the software or the information system itself. Others looked at the information system in the context of the organization and several looked at how specific information systems were used within work systems, although they did not use the term "work system." My personal belief is that IS research would benefit greatly from placing more emphasis on the relationship between specific information systems and the specific work systems they support. Keeping most of our attention riveted on the information systems per se reduces our ability to understand their operation in organizations and to interpret their significance.

User and stakeholder

Although several articles were quite clear about the importance of stakeholders other than users, most did not distinguish carefully between "user" and "stakeholder". In general, the term "user" was often ambiguous and might have meant anything from someone who enters data through someone who uses information through a manager in charge of work systems that use specific information systems. The articles generally did not consider the difference between information system usage and work system participation (by those users). Greater attention to this distinction might lead to more understanding of information system acceptance and usage.

Project

The projects mentioned in the papers ranged from software development projects (minus implementation in the organization) through organizational change projects that happened to involve information systems in some way (although the relative significance of the information system changes and other simultaneous changes may not have been clear). The distinction between development (achieving the desired software functioning) and implementation in the organization (achieving proper usage as part of a work system) was not always clear.

Implementation

The meanings of "implementation" ranged from programming and installing software that met requirements through getting the organization to use the software as part of a work system.

Reengineering

The meanings of "reengineering" ranged from modifying the internal operation of software to make it technically sound through substantially changing business processes in the organization.

Requirements

The meanings of “requirements” ranged from clearly documented processing capabilities that software should exhibit through general needs dictated by organizational change or competitive issues.

Solution

Use of “solution” to denote hardware and/or software sold by a vendor to address some purpose seems to have become part of the IS/IT vocabulary (unfortunately). This usage appeared in five of the ten papers.

FURTHER RESEARCH

This paper has explored some of the inconsistencies in the usage of basic IS/IT concepts, but it certainly has not plumbed the depths of this topic. Here are a few ways this research might be extended:

Similar Research, Different Articles or Issues

It might be valuable to perform a similar analysis on a different set of articles that are selected for a particular reason, such as to see whether articles from one perspective or another tend to express positive or negative feelings toward users or IT professionals. (See [Beath and Orlikowski, 1994].)

Make the Reader the Research Subject

Have different readers look at the same selected articles and look for systematic differences in what they perceive, in what they feel, or in the understanding they absorb. For example, characterize different readers as typically working from an IT perspective or a business perspective. Ask the readers to read descriptions of systems in operation or system projects. Look for the relationship between the reader’s perspective and the way the reader responds to the use of different perspectives in specific written examples.

Start with a Different Framework or Theory

This article used a particular theory of information systems as the basis of its attempt to characterize two perspectives and then find differences in the meaning of concepts based on these differences in the article's perspective. It might be interesting to perform a similar analysis starting from a different theory, or to compare the results of starting from two different theories of information systems. The main issue would involve the extent to which the theories affected the reader's understanding of a representative group of articles.

Test Whether Conceptual Confusions Matter

This article's underlying assumption is that inconsistent or contradictory uses of the same basic concepts matter. But look at the world. People can communicate about systems well enough to get their work done. Organizations do succeed in building information systems and in using them well enough that the majority are not abandoned. Even though the concepts are used inconsistently, we don't really seem to have a Tower of Babel on our hands. Perhaps everyday communication has enough redundancy that inconsistencies such as those discussed in this article are mostly a minor nuisance that wastes time but doesn't have major ramifications in practice. On the other hand, it might turn out that conceptual confusion leads to ineffective communication that becomes an obstacle to successful working relationships between IT professionals and business professionals.

V. CONCLUSION

So what? Ten articles published in *CAIS* between June and December 1999 seem to proceed from different perspectives (isn't that worthwhile?) and attach somewhat different meanings and connotations to common IS terms such as system, user, and implementation. Doesn't everyone know that IS is a multi-disciplinary field? Wouldn't this fact alone be reason enough to assume that authors would use some common terms differently? Isn't it too early in the development of this field to try to standardize terminology? And furthermore, aren't we smart enough to figure out what people actually mean from the context of what they say?

While it is too early to standardize terminology, it is worthwhile to recognize the problems that stem from attaching different meanings and connotations to the same words. We as a field seem terribly concerned with issues of rigor vs. relevance, as demonstrated by a 1999 issue of *MIS Quarterly* [Applegate, 1999; Benbasat and Zmud, 1999] and perennial panels on this topic at ICIS and other conferences. It is very hard to be rigorous with slippery concepts that legitimately mean different things to different people. (This is why [Seddon et al, 1999] looked at 186 articles to try to figure out what IS success means, and they came up with 30 different contexts.) It is also hard to be relevant with slippery concepts because communication is confused. That is where this article began: Sutter's letter [Sutter and Olfman, 1999] called for less user participation in IT projects. I initially thought he meant participation in IT projects by people who would use the information system directly; what I think he really meant was excessive participation in technical discussions by business executives who have little knowledge to contribute. Or maybe that isn't what he really meant.

And what about Olfman's call for additions to "the IS discipline's knowledge base"? Does the IS discipline really have a knowledge base? Assume that this knowledge base existed in ANY form ranging from some kind of oral tradition through a highly codified database of assertions along with supporting documents. It seems reasonable to argue that a knowledge base could not exist unless the basic concepts were fairly well defined. I won't go that far because we are clearly able to convey information to each other. I would say that greater clarity about basic concepts would probably make the accumulation and transmission of the knowledge much easier. This would increase rigor across the IS field (not just within self-referential articles) and would probably go far toward improving the relevance of our research.

Editor's Note: This paper was received on February 9, 2000. It was with the author for about 2 weeks for revision. It was published on April 21, 2000.

REFERENCES

Editor's Note: The following reference list contains hyperlinks to World Wide Web pages. Readers who have the ability to access the Web directly from their word processor or are reading the paper on the Web, can gain direct access to these linked references. Readers are warned, however, that

1. these links existed as of the date of publication but are not guaranteed to be working thereafter.
2. the contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. the authors of the Web pages, not CAIS, are responsible for the accuracy of their content.
4. the author of this article, not CAIS, is responsible for the accuracy of the URL and version information.

Alter, S. (1999a) "A General, yet Useful Theory of Information Systems," *Communications of AIS*, Vol. 1, Article 13, March 1999. http://cais.isworld.org/articles/1-13/

Alter, S. (1999b) *Information Systems: A Management Perspective*. 3rd ed. Upper Saddle Creek, NJ: Prentice-Hall, 1999

Applegate, L. M. (1999) "Rigor and Relevance in MIS Research – Introduction," *MIS Quarterly* (23:1), March 1999, pp. 1-3.

Beath, C. M. and W. Orlikowksi. (1994) "The Contradictory Structure of Systems Development Methodologies: Deconstructing the IS-User Relationship in *Information Engineering*" *Information Systems Research*, 5(4), December, 1994, 350-377

Benbasat, I. and R. W. Zmud. (1999) "Empirical Research in Information Systems: The Practice of Relevance." *MIS Quarterly* (23:1), March 1999, pp. 3-16.

Brodie, M.L. and Stonebraker, M. (1995) *Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach*, Morgan Kaufmann Publishers.

Coakes, E. and T. Elliman. (1999) "The Role of Stakeholders in Managing Change," *Communications of AIS*, Vol. 2, Article 4, July 1999. http://cais.isworld.org/articles/2-4/

DeLone, W.H., and E.R. McLean (1992) "Information systems success: the quest for the dependent variable", *Information Systems Research,* (3), pp. 60-95.

Giaglis, G. M. (1999) "On the Integrated Design and Evaluation of Business Processes and Information Systems," *Communications of AIS*, Vol. 2, Article 5, July 1999. http://cais.isworld.org/articles/2-5/

Gogan, J.L., J. Fedorowicz, and A. Rao. "Assessing Risks in Two Projects: A Strategic Opportunity and a Necessary Evil," *Communications of AIS*, Vol. 1, Article 15, May 1999. http://cais.isworld.org/articles/1-15/

Ives, B., & Olson, M. H. (1984). User involvement and MIS success: A review of research. *Management Science*, 30 (5), 586-603.

Jurison, J. (1999) "Software project Management: A Manager's View, " *Communications of AIS*, Vol. 2, Article 17, September 1999. http://cais.isworld.org./articles/2-17/

Kavakali, V. and P. Loucopoulos. (1999) "Modelling of Organisational Change Using the EKD Framework," *Communications of AIS*, Vol. 2, Article 6, July 1999. http://cais.isworld.org/articles/2-6/

Kelly, S., N. Gibson, C.P. Holland, and B. Light. (1999) "A Business Perspective on Legacy Information Systems," *Communications of AIS*, Vol. 2, Article 7, July 1999 http://cais.isworld.org/articles/2-7/

Lloyd, A.D., R. Dewar, and R. Pooley. (1999) "Legacy Information Systems and Business Process Change: A Patterns Perspective." *Communications of AIS*, Vol. 2, Article 24, December 1999. http://cais.isworld.org/articles/2-24/

Markus, M. L., and M. Keil. (1994) "If We Build It, They Will Come: Designing Information Systems That People Want to Use." *Sloan Management Review*, Summer 1994, pp. 11-25.

McFarlan, F.W., (1981) "Portfolio Approach to Information Systems," *Harvard Business Review*, Vol. 59, No. 5,pp. 142-150.

Metaxiotis, K.S., A. P. Papakonstantinou, J.E. Psarras. (1999): "Evaluating the Integrated Measurement and Evaluation System IMES: A Success Story," *Communications of AIS*, Vol 2, Article 15, September 1999. http://cais.isworld.org/articles/2-15/

Randall, D., J. Hughes, J. O'Brien, T. Rodden, M. Rouncefield, I. Sommerville, and P. Tolmie. (1999) "Banking on the Old Technology: Understanding the Organizational Context of 'Legacy' Issues," *Communications of AIS*, Vol. 2, Article 8, June 1999. http://cais.isworld.org/articles/2-8/

O'Callaghan, A. J. (1999) "Migrating Large-Scale Legacy Systems to Component-Based and Object Technology: The Evolution of a Pattern Language," *Communications of AIS*, Vol. 2, Article 3, July 1999. http://cais.isworld.org/articles/2-3/

Seddon, P. B., S. Staples, R. Patnayakuni, and M. Bowtell. (1999) "Dimensions of Information System Success," *Communications of AIS*, Vol. 2, Article 20, November 1999. http://cais.isworld.org./articles/2-20/

Sutter, J. and Olfman, L. (1999) "Letters to the Editor – 1: The Case of/for the Missing User," *Communications of AIS*, Vol. 2, Letters 1, Nov. 1999. http://cais.isworld.org/letters/2-1/

APPENDIX I: BASIC CONCEPTS ABOUT WORK SYSTEMS, INFORMATION SYSTEMS, PROJECTS, AND USERS

A *CAIS* article entitled "A General, Yet Useful Theory of Information Systems" [Alter, 1999a] used the concept of work system as the basis of an integrated view of topics such as system, information system, project, and user. Since the CAIS article is readily available to readers of this article, the main points related to systems, system projects, and users are summarized below with very little embellishment. Although they take a business perspective they are not about business strategy or measures of business success. Instead they are meant to be the basis of an operational description of what systems are and how they operate in organizations. The concepts start with "work system" because this is a common denominator that applies across information systems, projects, and other systems that don't use computers at all.

The numbers in parenthesis, below, refer to one of the 14 numbered points in [Alter, 1999a].

WORK SYSTEMS AND INFORMATION SYSTEMS

A work system is a system in which human participants and/or machines perform a business process using information, technology, and other resources to produce products and/or services for internal or external customers. Organizations typically contain multiple work systems and operate through them.

(1) Understanding a work system therefore requires at least cursory understanding of six elements: the business process, participants, information, technology, products, and customers. (2)

An information system is a particular type of work system. An information system is a work system whose internal functions are limited to processing information by performing six types of operations: capturing, transmitting, storing, retrieving, manipulating, and displaying information. An information system exists to produce information and/or to support or automate the work performed by other work systems. Information systems may serve other work systems through a variety of roles. (5) In relation to a single work system, an information system may provide information for decision making, may structure or control the work, or may automate some of the work. In relation to a group of related work systems, an information system may support information sharing, may coordinate the work, and may integrate the work. (6) The integration between an information system and a work system can take on many different forms. The information system may serve as an external source of information; it may be an interactive tool; it may be an integral component of the work system; the information system and work system may overlap so much that they are virtually indistinguishable. The information system may also serve as shared infrastructure used in many diverse work systems. (7)

The definition of information system is important for understanding much of the IS literature because research findings may be stated as though they apply to information systems in general even though they implicitly refer to particular types of information systems such as transaction processing systems, management information systems, or communication systems. Conversely, generalizations, truisms, and success factors related to work systems in general should also apply to information systems and to projects, just as generalizations about information systems and projects should apply to specific types of information systems and specific types of projects. (14)

An information system can be viewed as consisting of content and plumbing. Its content is the information it provides and the way that information affects the business process within the work system. Its plumbing is the details that concern information technology rather than the way information affects the business process. In principle, plumbing should be hidden from work system participants to the extent possible. (8) This implies that information system participants and information users should be involved in defining the content of the information system, but may not have much to contribute in designing its plumbing. The reverse caveat applies to IT professionals. The fact that IT professionals know a great deal about an information system's plumbing may imply little about their understanding of the information system's content and the role its content plays in work systems it supports.

Software is part of the technology in an information system. Software defines the data and the methods the computer uses for processing data. Part of the software codifies the content, but the software itself is part of the plumbing.

PROJECTS

A project is a time-limited work system designed to produce a particular product and then go out of existence. (10) A typical project designed to change a work system or an information system is broader in scope than a software project because it includes changing the way people do their work rather than just changing the way software operates on a computer.

Phases of a Project

Regardless of whether an information system is involved, a project that creates or significantly changes a work system goes through four idealized phases: initiation, development, implementation, and operation and maintenance. (12) When an information system is involved, the same phases apply regardless of whether the information system is built from scratch using a structured life cycle, is based on application software purchased from a software vendor, or is developed using a sequence of prototypes. An information system textbook

[Alter, 1999b] shows how the phases map to different system development methods.

Initiation is the process of clarifying the reasons for changing the work system, identifying the people and processes that will be affected, describing in general terms what the changes will entail, and allocating the time and other resources necessary to accomplish the change.

Development is the process of defining, creating, or obtaining the tools, documentation, procedures, facilities, and any other physical and information resources that are needed before the change can be implemented successfully in the organization. When changes in an information system's content are involved, development starts by creating detailed specifications of what the information system' s content will be or how it will change. Business professionals should play an important part in defining these requirements even though they do not participate in other development activities such as internal design and programming.

Implementation is the process of making the desired changes operational in the organization. This includes planning for the roll out, training work system participants, and converting from the old way of doing things to the new way.

Operation and maintenance involves keeping the work system operating effectively by monitoring its performance and making minor changes that do not require a major project. This phase continues until major changes are required and a new iteration of the four phases starts.

Software Project vs. Information System Project vs. Work System Project

The difference between a software project, an information system project, and a work system project is noteworthy because work system participants have much smaller roles in software projects than in information system projects or work system projects. For example, since plumbing should be hidden from work system participants to the extent possible (8), projects that are mostly about plumbing should have comparatively little or possibly even no user involvement at all. (This is a restatement of a main point in [Sutter and Olfman, 1999].) In contrast, business professionals should be involved in all the major phases of an information system project that changes information system content to a significant degree. They should control content, either directly or indirectly, because they have the best understanding of how the work system should operate. Their participation is also needed because it usually gives the project additional credibility that removes obstacles during implementation.

Software project is a time-limited work system whose goal is to produce software that meets a particular requirement. Y2K remediation is a good example of a software project. The project starts with some existing software and its goal is to remove flaws from that the software without changing its intended function. The goal has nothing to do with changes in the external world in which that software is used. A new release of a word processing program is another example of a software project (rather than an information system project) because the direct result is shrink-wrapped or downloadable software rather than changes in an operational information system in a particular organization. The project manager for a software project declares victory when the software runs on the computer in the desired manner, regardless of whether it is being used effectively by anyone. A software project may be part of a larger information system or work system project. If so, the overall effort is not a complete success until the information system or work system is operating as intended.

Information system project is a time-limited work system whose goal is to create or modify an information system so that it operates in accordance with a set of requirements and is maintainable. An example of an information system project is building a new tracking system for sales. The essence of the work system is doing the selling, but the information system supports the work system in many ways. The project manager declares victory when the information system is operating in the desired manner, whether or not the sales work is being done more effectively.

Work system project: is a time-limited work system whose goal is to create or modify an operational work system so that it operates in accordance with a set of requirements and is maintainable. An example of a work system project is creating a new way to perform sales work. This often involves a new information system that is needed in order to do the work in a new way. The project manager declares victory when the sales work is being done in the desired manner.

Work System Elements for Different Types of Projects

Since a project is a work system and since understanding a work system requires at least cursory understanding of six elements (2), some of the differences between software projects, information system projects, and work system projects can be appreciated in more depth by looking at the six elements.

Customer

Projects within the general realm of software, information systems, and work systems can have many different types of customers:

- hands-on users of the software or the information systems,
- users of the information produced by the information system, whether or not they are hands-on users of the software or information system,
- managers of the people who use the software or the information produced by the information system,
- IT professionals who must maintain the software over time.

The customers of projects that are mostly about IS plumbing and have little visibility to users tend to be the IT professionals. The customers of projects that change IS content may include any of the above.

One of the main issues in customer satisfaction is whether the modified or newly created software, information system, or work system satisfies the goals of set or perceived by the various customers, who often disagree about what they want. The hands-on users may want functionality that is more convenient and fits better with their other work system responsibilities. The business managers may want anything from greater flexibility relative to the external environment through lower charges against their budgets. The IT professionals may want software and hardware that is easier to maintain. With so many types of concerns, it is quite possible for a software or information system project to meet some or all of its requirements and still not satisfy some of its customers.

Product

The product is quite different across the three types of projects. For a software project the product is new or modified software that satisfies requirements. For an information system project it is an information system that meets requirements and is operational in the organization. For a work system project it is a work system that operates as intended. IT professionals have direct control over the software they build but have little or no control over the work systems they are trying to support. Consequently, they have much more control over the success of a software project than they have over the success of a content-related information system project or a work system project.

Business process

A project that creates or significantly changes a work system goes through four idealized phases: initiation, development, implementation, and operation and maintenance. (12) When an information system is involved, the same phases apply regardless of whether the information system is built from scratch using a structured life cycle, is based on application software purchased from a software vendor, or is developed using a sequence of prototypes. In contrast, a software project may end after the development phase because the goal is to produce the software rather than to change a particular organization.

The meaning and significance of implementation is the most interesting topic in comparing work system projects, information system projects, and software projects. In the four phases of a work system project, implementation is about converting from the old information system and work system to the modified information system and work system. The literature about software projects often implies a different meaning for the term implementation. This meaning involves satisfying a set of functional requirements for software. In the four phases of a work system project, implementation ends when the new systems are operational in the organization. The alternative view is that implementation ends when the software operates correctly on the computer. This difference in terminology would be of little consequence except that similar confusions frequently muddle project-related discussions between business professionals and IT professionals. To say the least, the fact that implementation may have totally different meanings to people participating the same project is likely to cause confusion. (10)

Participants

The active participants in software projects are mostly IT professionals, although business professionals may provide input related to requirements. The participants in information system projects and work system projects include both IT professionals and business professionals. At any given size, projects that change both information system content and information system plumbing are probably more difficult than projects that just change plumbing or content. (12) This is because these projects call for a wider range of participants with a wider range of goals, interests, and professional affiliations.

Information

Work system and information system projects involve a wider range of information than software projects of a comparable size because more factors must be considered and because the business process of performing the project is more extensive, especially in the implementation phase.

Technology

Since information system and work system projects usually include software projects there is no general difference in the technology used in these projects.

Failure Modes for Projects

The three different types of projects have different failure modes. Failure of an information system project may occur due to failure of a software project that it includes, or due to other causes unrelated to the software project, such as changes in business conditions. Similarly, failure of a work system project may occur due to failure of an information system project that it includes, or due to other causes unrelated to the information system project.

Software project failure. Some software projects are never completed, such as the American Airlines Confirm System. Other software project failures occur in the form of catastrophic bugs that become apparent after the project seems to be complete. Examples include the software failure in the Mars Orbiter caused by inconsistent use of English and metric measures, the programming bug that crashed the AT&T phone network several times in the 1990s, and the guidance system bug that caused the destruction of an Ariane 5 rocket.

Information system project failure may occur due to software project failure or for other reasons unrelated to a software project. Where the problem is not a software project failure, the software meets requirements but the information system is not used effectively. An example is presented in [Markus and Keil, 1994], which tells the story of an information system that was not used effectively despite being redesigned to satisfy to agreed upon specifications. In this case the specifications did not adequately reflect the reality of the work system in which the users were participants. Other examples of this type are information systems designed to support information sharing but not used effectively because too few people really wanted to share information. Troubled implementations of ERP systems fall into this category as well.

Work system failure. Work system projects may fail because of an information system project failure or for other reasons unrelated to an information system project, such as employee turnover, internal political obstacles, insufficient resources, general mismanagement, the organization's inability to attract the right employees, and the organization's inability to adjust to competition and changes in the external environment.

APPENDIX II: REPRESENTATIVE STATEMENTS FROM TEN *CAIS* ARTICLES

Table A.1: Statements providing evidence for an IT perspective vs. business perspective in Vol. 2, Article 3, "Migrating Large-Scale Legacy Systems to Component-Based and Object Technology: The Evolution of a Pattern Language," (O'Callaghan, 1999)

Topic	*Statements from the article*
System	• This paper presents a successful new approach which focuses primarily on the architecture of the software system to migrate an existing system to a new form. • Prioritization of requirements is typically dictated, *in the final analysis*, by the business context the software system serves, but in the first analysis there are usually a host of hidden assumptions underlying the business perspective itself. • Expertise in shifting legacy systems to new paradigms is buried in the folklore of software engineering. • However, the movement of any large-scale business-critical system to components is fraught with difficulty. • Legacy systems have been defined as stand-alone applications built during a prior era's technology but they are perhaps more widely understood as software systems whose plans and documentation are either poor or non-existent.. A more useful definition … is: "A legacy system is a large system delivering significant business value today from a substantial pre-investment in hardware and software that may be many years old. … It is a business-critical system which has an architecture which makes it insufficiently flexible to meet the challenges of anticipated future change requirements" • Meanwhile successful systems have simply aged, some less gracefully than others. Jones estimates that the average rate of change of software systems is between 5% and 7% every year, year on year.

	• Mergers, takeovers, shutdowns and corporate restructuring can turn well-planned, well-engineered up-to-date systems into obsolescence virtually overnight. • "How do we best support the sale of new financial products, and what is the optimum configuration of IT for this purpose?". The solution may indeed involve new software development to replace the old system, or it may mean simple, incremental enhancement of the old system, or a mixture of both. The point is that the legacy problem is primarily a business problem, and only incidentally a technical one, and that any solution must be driven from the problem space.
User	• …The development team was able to capture the essence of the business problem by working with the users. • One of Alexander's first contributions was to reject the modern split between architect (who theorizes) and builder (who constructs, following the architect's drawings) in favour of a combination of user (inhabitant)-centred design and an architect-builder model in which the architect also implements
Stakeholder	(This term is not used, but the term user seems to be a synonym in this paper.)
IS project	• The *raison d'être* for contemplating a move to an object-based representation for an existing system is the belief that business benefits in terms of increased flexibility to business change, and increased productivity (through software reuse) will result. •All the projects were considered to be successful in terms of their immediate technical objectives, their medium to long-term business objectives, and in their strategic and tactical research objectives. • Legacy information systems are typically the targets of reverse engineering projects. • Best practice is captured in the form of software patterns that address not only the design, but crucially also the process and organizational issues that inevitably surround such a project. • The business case [is] the key criteria for determining whether or not to migrate a legacy system and …the software architecture [is] the main focus of attention for the migration process.
Reengineering	• The migration of legacy systems is a process of re-engineering. The accepted definition of re-engineering is … "the examination and alteration of the target system to reconstitute it in a new form". • There is a particular quality to the re-engineering effort that must be understood when it involves moving a computer system from, say, a structured representation to an object-based one, however. • The relative failure of traditional reverse engineering techniques when applied to the restructuring of systems to an object-based or object-oriented form results from their tendency to ignore the changing problem space which, typically, is driving the need for change in the first place.
Implementation	• When a shift is being contemplated from, say, representation in a structured language to representation in an object-oriented implementation, it is not just the language that is changing but the development paradigm itself.

	• The architecture is did *not* necessarily imply an object-oriented *implementation*. Indeed, the first two pioneering projects delivered a restructured system in the same base technology in which the legacy system was originally implemented. • Irrespective of the target implementation technology, object modelling was used to capture a description of the existing system in its business context, describe the new architecture, and plan the technical migration. • The next steps largely concern the detail of the implementation abstractions, which will, of course, include legacy code. • This scoping of the analysis model so that it captured the key abstractions of the problem space and modelled them separately and independently of any implementation concerns reflects the *Shamrock* pattern of the ADAPTOR language. • By postponing consideration of the *representation* of the key VAT abstractions in software, and of their implementation and interfacing to other components in the customer service system, the development team was able to capture the essence of the business problem by working with the users. • The basic notion is that classes which exist to access legacy code should differ from other objects only in their implementation details.
Requirements	• [A legacy system] is a business-critical system which has an architecture which makes it insufficiently flexible to meet the challenges of anticipated future change requirements. • A software system that tries to meet the requirements of all possible scenarios will almost certainly suffer 'analysis paralysis' and will be too complicated and/or inefficient to deliver and use. • A software architecture that maps closely onto the key abstractions …[increases] the likelihood of maintaining traceability from solutions to requirements through such business-driven changes. • In each case the systems' owners made a business decision that they needed a component-based architecture in order to meet the challenge of ever new requirements, and this architecture implied the kind of encapsulation that object-based systems deliver. • This understanding frees the developer to utilize the same requirements gathering and modelling techniques to describe any part of a system that could be used to describe the system as a whole. • The utilization of use cases to capture the 'as is' requirements reflects pattern 22, *Scenarios Define Problem* in Coplien's organization and process pattern language.
Solution	• The spread of the PC from the mid-1980s encouraged a culture in which 'point solutions' were developed. • Similar applications running on different operating systems on different boxes became common. Worse still, key business abstractions such as 'Customer' could be running on different applications on the same machine at the same time, and since these applications could not talk to each other, information integrity could not be maintained. Subsequently, such point solutions became subject to localized optimizations, and uncontrolled maintenance, etc., exacerbating the position even

	further • But these benefits rely, as we have seen, on the fact that object systems 'break' from the underlying Von Neuman architecture of the machine and enable the possibility of building software solutions in the image of the problem space itself. • A software architecture that maps closely onto the key abstractions …[increases] the likelihood of maintaining traceability from solutions to requirements through such business-driven changes.
Point of reference	• (Explaining the use of a pattern language in software migration projects) The experience of four successful migration projects in five years has clearly demonstrated clearly the importance of focusing on software architecture -–(the partitioning of a system according to a specific separation of concerns -) and on achieving a strong correspondence between the key abstractions in the problem space and software components in the solution space.

Table A.2: Statements providing evidence for an IT perspective vs. business perspective in Vol. 2, Article 4, “The Role of Stakeholders in Managing Change” (Coakes and Elliman 1999)

Topic	*Statements from the article*
System	• A Computer Information System (CIS) …faces continual redevelopment to respond to the changing organisational needs. …Change can only be effective if the plans recognise those who have a stake in the process and they are led to see the value in the new structures or systems. • In establishing a definition appropriate for CIS development it is necessary to consider notions of the system boundary and influences from outside a formal organisation. • Development of new or modified systems in the presence of legacy systems is normal for most companies. Today's new system will become the legacy system in the next, inevitable, round of change. • Change can only be effective if the plans recognise those who have a stake in the process and they are led to see the value in the new structures or systems.
User	• Computer Information Systems development often focuses on direct users and affected internal departments as the exclusive stakeholders. However these groups may present too narrow a perspective. • Looking beyond the immediate users of the system greatly increases the number of views which need to be addressed.
Stakeholder	• For our purposes a stakeholder is someone who has an interest in a CIS development *and* can affect the success of that development. • A particular risk is that close to the technical boundary we will find stakeholders who have extreme views of existing, or legacy systems. The danger is that inappropriate factors may be given more weight than the wider needs of the organisation and its environment. Stakeholders close to the technology can be expected to express their personal investment in the current technology, their detailed experience of operational problems, or their technological bias lending enthusiasm for the promises of new

	technology.
IS project	• Computer Information Systems development often focuses on direct users and affected internal departments as the exclusive stakeholders. However these groups may present too narrow a perspective. To improve the effectiveness of the development process, a wider constituency should be considered that includes organisational partners in the wider business environment. • [The following themes] capture a pragmatic dimension to justifying a "stakeholder's" participation. - Stakeholders may affect realisation or may be affected by realisation of a system; - Stakeholders may have actual versus legitimate influence; they may be an internal affect or versus external; - Stakeholders may have a supportive influence versus conflictive influence - They may be stakeholders of a common value; they therefore need to be considered, consulted, participative, or responsible for process under consideration or development by the system.
Implementation	• Another failing in the group's decision making was the difficulty in separating technology and implementation from strategic decision making.
Reengineering	(Not discussed in the article)
Requirements	• [In relation to a case study] Many of these external stakeholders are governmental bodies whose needs for the supply of data and reports are an integral requirement of any university's student record keeping package. • [Paul] identifies six environmental changes that have the potential to affect an organisation's CIS needs. [The first is] changes in legal requirements.
Solution	• Initial planning was at a level of detail that had to be discarded when the emergence of Internet technology and applications provided a readily implementable solution in 1995
Point of reference	• This paper presents a method, the stakeholder web that identifies appropriate stakeholders and their viewpoints.

Table A.3: Statements providing evidence for an IT perspective vs. business perspective in Vol. 2, Article 5, “On the Integrated Design and Evaluation of Business Processes and Information Systems” (Giaglis, 1999)

Topic	*Statements from the article*
System	• The role of information systems in influencing and enabling organisational design is widely acknowledged. Yet limited attention is paid to the theoretical legitimacy and conceptual basis of IS-enabled organisational change • Most modern change management approaches differentiate from their older

	counterparts by their focus on the business process as the fundamental unit of analysis in organisational design. According to the perspective they advocate, organisations should not be analysed in terms of the functions into which they can be decomposed or in terms of the products they produce, but in terms of the key business processes that they perform. • Contemporary IS are increasingly integrated together, making it even more difficult to disentangle a single system for evaluation. This may render the demarcation of boundaries around individual systems for the purposes of evaluation a meaningless exercise
User	• Many systems still fail to fulfil the needs of their users and the organisations that adopt them.
Stakeholder	• Project champions tend to underestimate costs and overestimate benefits. • Interviews with key process participants (management and employees) of both companies were conducted to capture the process essence and decompose the order fulfillment process into its component activities. The knowledge elicited by the interviews was used to define the boundaries of the process and the models to be developed.
IS project	• SDLC-based IS development methods …perpetuate the distinction between the business and the IS domain. Most structured approaches to IS development begin with an implicit assumption that the business domain issues are resolved and the system is to work in a stable and well-defined business environment, where the only issue is to identify the 'correct' requirements for the new IS. As a result, not enough attention is generally being paid to investigating the interactions of the IS to be developed with the business processes it will naturally affect. • The design and implementation of information systems is generally a complex and laborious exercise for most contemporary organisations. It may not be desirable (or even feasible) to incorporate such design into business process change in its entirety. A strategy where IS design is treated along two dimensions (one concerning the organisational impact of IS, and the other concerning the technical implementation details) may be more appropriate.
Reengineering	• Business engineering is defined here as the integral, concurrent design of organisational processes and the information systems to support them. • The challenge for business engineering is to bring process design and IS design together without adding to the already high complexity of each task alone. One way to achieve unity is to incorporate high-level IS design into business process design projects and leave the technical details of IS implementation to be managed in the aftermath of process change decisions.
Implementation	• This point reinforces our earlier argument for incorporating only the high-level organisational impacts of IS in business process design and leaving the low-level technical implementation details for later. • Finally, SDLC-based approaches tend to view IS evaluation as a post-implementation activity, addressed only in the last step of the system development life cycle. • What may be needed is an explicit focus on the pre-implementation (*ex ante*) evaluation of the information system (for example, within the problem identification or system analysis stages).

	• [IS] evaluation provides the benchmarks of what is to be achieved by the IS investment. These benchmarks can later be used to provide a measure of the actual implementation success of IS projects. • Further to the simulation analysis, the process scenarios were scrutinised to develop a detailed understanding of implementation challenges and transform hypotheses into detailed implementation plans. The requirements of each option regarding technology, people, and skills were assessed and a formal cost-benefit analysis was conducted to evaluate the proposed investments. • The challenge for business engineering is to bring process design and IS design together without adding to the high complexity of each task alone. … A potential strategy for addressing this need would involve incorporating high-level IS design and IS evaluation into business process design, and leaving the technical details of IS implementation to be addressed in the aftermath of business engineering decisions. Such an approach was followed in the case study where the EDI applications were defined in general terms (only to the level of detail necessary for the model development and analysis) without the need for specific reference to implementation-dependent technical details. What is even more important is that implementation details need only be developed for the solution chosen and not for every alternative information system design that was considered during the business engineering endeavour. • The design and implementation of information systems is generally a complex and laborious exercise for most contemporary organisations. …A strategy where IS design is treated along two dimensions (one concerning the organisational impact of IS, and the other concerning the technical implementation details) may be more appropriate. • Such methodologies should satisfy the requirements identified above, namely adopting a process perspective in analysing organisational structures, integrating high-level IS design within business process design, and leaving the technical details of IS implementation to the software engineering domain experts.
Requirements	• Most structured approaches to IS development begin with an implicit assumption that the business domain issues are resolved and the system is to work in a stable and well-defined business environment, where the only issue is to identify the 'correct' requirements for the new IS. • The life span of IS is uncertain (due to technological obsolescence and changing requirements). • The requirements of each option regarding technology, people, and skills were assessed and a formal cost-benefit analysis was conducted to evaluate the proposed investments. • ..Although most existing IS development methods begin by stressing the importance of understanding the real-world operation that the IS will support, they quickly become absorbed in the definition of individual functions and detailed requirements ('reductionism').
Solution	• In line with the previous analysis, business process simulation was employed to assist in identifying the problems of existing process designs, to formulate appropriate solutions based on EDI applications, and to realise the expected impacts of these solutions on key business performance indicators.

	• Simulation made it possible to realise that, if combined with the technology introduction, other (non EDI-dependent) structural process changes could provide a solution to the inefficiencies of the process.
Point of reference	• We are aware of no IS evaluation method that actually advocates such a perspective [using the business process as the unit of analysis] for appraising the benefits of an information system by measuring the impact of changes on the level of the business processes that the IS is intended to support. • We need to adopt process change as a mediating factor between the IS initiative and economic return. Such thinking could trigger a radically different perspective in the way IS investments are viewed and analysed within an organisation.

Table A.4: Statements providing evidence for an IT perspective vs. business perspective in Vol. 2, Article 6, "Modelling of Organisational Change Using the EKD Framework," (Kavakali and Loucopoulos 1999).

Topic	*Statements from the article*
System	• While information systems continue to serve traditional business needs such as co-ordination of production and enhancements of services offered, a new and important role has emerged: the potential for such systems to adopt a supervisory and strategic support role. • Any type of change whether it involves the development of a computerised system or the re-engineering of business processes involves many assumptions about the embedding enterprise domain. • Prior to designing new business processes and support information systems, any reform requires a clear understanding (and a sharing of this understanding among many stakeholders) of the current enterprise situation. •
User	• [mentions users of the EKD framework, as in:] -The EKD roadmap is a navigational structure in the sense that it allows the roadmap user to determine their route between the different knowledge states regarding organisational change. - For example, if the user has no knowledge about the organisation then the entry point will be the Null state. - Thus, the electronic roadmap can be used by different users at different sites in the organisation.
Stakeholder	• The implications of these forces on this organization [part of an electricity company] is that, prior to designing new business processes and support information systems, any reform requires a clear understanding (and a sharing of this understanding among many stakeholders). • Both scenario evaluation as well as interpretation of evaluation data was dependent on subjective judgement of involved participants. Finally, it should be noted that the evaluation data provided organisational stakeholders with a rationale means of making an informed choice.
IS project	• Change goals form the requirements upon which the re-engineered enterprise

	structure will be based. This task concerns the mapping of change requirements onto a future enterprise model, which in turn involves the modelling of the future enterprise goals and how these goals will be realised in terms of operational enterprise components. • The EKD framework for modelling organisational change defines the set of applicable knowledge states that need to be reached in an organisational change project. However, it does not dictate any particular ordering between these states…. Instead, each state to be reached is dynamically selected in the course of the change management process. Each route characterises a specific method for solving the problem at hand.
Implementation	• The implementation comprises introducing customers profiling, minimising delay time to serve an application, offering all means for payment, offering all services at customer premises, introducing all available technologies to communicate with customers, introducing IT solutions for all services.
Reengineering	• Any type of change whether it involves the development of a computerised system or the re-engineering of business processes involves many assumptions about the embedding enterprise domain. • In a business process re-engineering project, one may start by understanding the current situation (reach the As-Is state) and proceed with exploring alternative change scenarios (reach the Change state), continuing with the evaluation of alternative scenarios (reach the Evaluation state) and finally, design the re-engineered business processes according to the selected change plan (reach the To-Be state).
Requirements	• Change goals form the requirements upon which the re-engineered enterprise structure will be based. This task concerns the mapping of change requirements onto a future enterprise model, which in turn involves the modelling of the future enterprise goals and how these goals will be realised in terms of operational enterprise components. • Using the EKD ends-means links, change in enterprise goals (regarding for example, company objectives, policy, general market condition) will propagate top-down as reasons or requirements for re-organising the enterprise processes. • These sessions resulted in the specification of both internal enterprise needs as well as external constraints that defined the enterprise change requirements. • GroupSystems is a suite of team-based decision software tools that were used for the identification, elaboration and resolution of stakeholder requirements. • Having agreed on a set of change requirements the next step in our route was to identify how these requirements could be compared and contrasted with the current goals, thus providing a basis for a reasoned approach for future improvement. This task resulted in the identification of alternative change scenarios indicating the type of organisational transformation necessary for satisfying change requirements.
Solution	• The aim of evaluation is to deliver an enterprise model, which is consistent with the stakeholders' experience and/or expectations. Often, alternative enterprise models may be possible (e.g., there may be multiple change models, leading to alternative future solutions). • This approach focuses on the systematic analysis of the effects of change requirements on the existing enterprise context, rather than prescribing a solution based on experts' opinions
Point of reference	• Modelling of organisational change in EKD is achieved through the use of: a common

	set of concepts for describing enterprise knowledge regarding organisational change, i.e., the EKD enterprise ontology and a methodology roadmap and associated guidelines for assisting user navigation within the space of the possible routes connecting the four knowledge states (As-is, Change, To-Be, and Evaluation).

Table A.5: Statements providing evidence for an IT perspective vs. business perspective in Vol. 2, Article 7, "A Business Perspective on Legacy Information Systems," (Kelly et al, 1999)

Topic	*Statements from the article*
System	• Legacy information systems can represent a huge investment for organizations in terms of information technology, business processes, procedures and organizational structures. • Legacy information systems include business and technical dimensions … the systems can present problems when there is a misalignment between the strategic vision of the business, the IT legacy and the old business model embodied in the legacy. • Legacy information systems are defined as information technology (e.g. hardware, software applications and network) and the business model implicit in the application of that technology (e.g. organizational structure, work flows, procedures and processes) within the organization. • First-generation systems dating from the 1960s and 1970s were in machine language but most were developed in assembly or early versions of third-generation programming languages such as COBOL or FORTRAN • Second-generation systems (late 1970s and throughout the 1980s) possessed some degree of modularity and many were used for online transaction processing. •The real value to the organization of information technology legacy systems lies in the "accumulation of years of business rules, policies, expertise and 'knowhow' embedded in the system. • The business legacy is embedded in the legacy IT system, and it is the inter-relatedness of business and IT legacy which makes either business or technical change a difficult process. • Interdependence can make it difficult to predict changes arising in the system as a whole as a result of minor enhancements to one component.
User	(Not used in significant way in article.)
Stakeholder	• The inclination and acceptance of change does not exist within the culture of the organization and hence employees resist change. Although managers see change as an opportunity to strengthen the business, employees may perceive change as disruptive and intrusive.

IS project	• Legacy systems comprise the existing characteristics of an organization such as its structures, processes, strategy and cultures resulting from the impact of internal and external forces. These characteristics can mean that even when new emerging technologies are introduced into organizations, employees have difficulty in adapting to new ways of working.
Implementation	• The existence of legacy systems that hinder the implementation of new business strategies is now well established. • The company decided to reengineer the organization including implementation of a process-oriented ERP solution.
Reengineering	• Many organizations are finding that legacy information systems act as a barrier to strategic innovation. • The gap between what the legacy information systems can deliver and the strategic vision of the organization widens when the legacy information systems are unable to adapt to meet the new requirements.
Requirements	• In addition to date and regulation requirements, a range of business pressures are increasingly significant today •The information systems were not integrated between sites or within sites. They could not support the MIS requirements for a profit oriented business. i.e. measure profitability and monitor operating costs
Solution	• … as technology moves beyond traditional transaction processing towards client/server architectures and the Internet to create new types of business solutions. • The company decided to reengineer the organization including implementation of a process-oriented ERP solution. • …decided to move towards a process-oriented approach facilitated by an ERP solution.
Point of reference	• Legacy information systems are usually considered from a technical perspective, addressing issues such as age, complexity, maintainability, design and technology. We wish to demonstrate that the business dimension to legacy information systems, represented by the organisation structure, business processes and procedures that are bound up in the design and operation of the existing IT systems, is also significant.

Table A.6: Statements providing evidence for an IT perspective vs. business perspective in Vol. 2, Article 8, "Banking on the Old Technology: Understanding the Organizational Context of 'Legacy' Issues," (Randall et al, 1999)

Topic	*Statements from the article*
System	• 'Legacy', we argue, is not just a problem encountered by organisations with aging mainframes and dated software, it is an issue from the moment a computer system becomes an integral part of any organisation's everyday work. • 'The system as a whole, and both main software packages, were seen as 'dated', 'slow', prone to unpredictable breakdown, and not 'user friendly'.

	• The main 'workhorse' systems in use in the bank were two software packages: BAF, an accounting/bookkeeping package dating from the 1960s, that, "had bits bolted onto it"; and ISS, a more modern relational database • Financial institutions were among the first wave of business organisations to computerise many of their operations. A great deal of their basic functioning is now dependent on those aging systems. • You've got to use it as a tool…using the software to confirm rather than determine decisions … may have arisen as a consequence of the inclusion in the program of 'non-financial' information which could significantly influence the risk grade obtained. • Workers were required to indicate that they had completed all the formalities on each screen before they would be permitted (by the machine) to proceed to the next. This rigid workflow model would, however, occasionally create problems… there were occasions when they needed to subvert the strict workflow model.
User	• Computer systems have been installed in many companies for some time now and no matter how well they may have fitted the situation initially, usage and the circumstances of use have changed, as indeed have the needs and the users, and, most importantly, the organisations themselves • Observation and conversations with users indicated a number of problems. The system as a whole, and both main software packages, were seen as 'dated', 'slow', prone to unpredictable breakdown, and not 'user friendly'.
Stakeholder	(Not mentioned explicitly. Mentioned implicitly in statements about users and about IS projects)
IS project	• It is unlikely that any organisation is ever 'going to get it right' the first time. What it *does* suggest is the need for more effective monitoring of new technologies in their situations of use and developing effective mechanisms for involving users' experiences in development. • No matter how promising new technologies may seem, the realities of their implementation are typically disruptive. They also involve huge overheads in respect of retraining and compensatory payments, not to mention the lead-times required for familiarisation of workers using the system. These problems arise whenever new systems are introduced, no matter how carefully the planning was done. • Part of the 'centralising' objective was an attempt to ensure that, for every single process in which the bank engaged, there would be a process map so that anyone could come in and do the job in exactly the same way as anyone else. … it was necessary for workers involved in different aspects of the lending process to arrive at some sort of understanding of the work of others involved in the same process, beyond their own teams, and sometimes beyond the walls of the Lending Centre itself.
Implementation	• No matter how promising new technologies may seem, the realities of their implementation are typically disruptive.
Reengineering	(The term reengineering is not used, but this seems to capture the authors' view of reengineering:.) • "…. Apparently small changes may have major implications." This statement is not necessarily an indication of the unwillingness of those responsible for the development

	of the system to make appropriate changes. It is equally likely to be an indication of just how difficult it is to modify systems which are already in use and upon which the work depends, not to mention the problems of technical complexity. In significant respects, problems such as these are as much organisational as technological because they direct attention to the need to reorganise work and implement new technologies in a more integrated way.
Requirements	• The two managers had to work together to produce some kind of model that seemed to give due consideration to their own, highly particular requirements. The end product was a complex and highly creative design that was heavily informed by their own experience of the day-to-day character of their work, and the work of the staff around them.
Solution	• Ethnographic methods … bring a particular focus to the analysis of systems in use and thereby outline the 'play of possibilities' for work and design, "enabling designers to question the taken-for-granted assumptions embedded in the conventional problem-solution-design framework."
Point of reference	• Legacy concerns are not merely technological in focus but also organizational in the sense of being intimately wrapped up with the everyday accomplishment of work. • The rapidly changing nature of commercial and organisational life means that legacy issues can arise relatively soon after the introduction of comparatively new technologies. Moreover it would seem that that an appreciation of legacy needs to move away from a purely technological stance to admit the importance and impact of organisational issues. • Straightforward process approaches, despite their attraction to system modelers, are unlikely to take into account the various interactional subtleties involved in the actual doing of the work. In that case understanding how 'processes' may be made efficient and effective would seem to require a nuanced view of various factors, including working practice, communication and control problems, and indeed any number of complex articulations of structure, process, technology, and 'situated' knowledge.

Table A.7: Statements providing evidence for an IT perspective vs. business perspective in Vol. 2, Article 15, "Evaluating the Integrated Measurement and Evaluation System IMES: A Success Story" (Metaxiotis, Papakonstantinou, Psarras,1999)

Topic	*Statements from the article*
System	• IMES is an integrated information system that incorporates Internet technologies to provide wide monitoring and evaluation capabilities. It consists of five individual, but interacting, subsystems that form a robust intranet information system. The subsystems are the database (thousands of management reports from 1994 to 1999), the local application, the input/output assistants, the intranet component, and the security mechanism. • The system provides management information on project implementation, so that structured management decisions can be taken. • The VB Script language, which creates these Web pages, submits calls to the system

	database using ODBC driver technologies. • This database is the "back end" application used for storing all kinds of data. It is built in MS Access 7.0 and structured according to the relational model into entities and relationships.
User	• To most users, the interface is the system. • Effective interfaces do not concern the user with the inner workings of the system • [The information system] [provides relevant data to the users] and provides [search capabilities]. • [Evaluation criteria include:] - Reliability: The extent to which the clients can trust the system and its services - Accessibility: The degree to which the system database is easy to be accessed by the users. - Ease of use: The extent to which the users can "navigate" in the system database and use its services.
Stakeholder	(Stakeholders other than direct users were not mentioned directly.)
IS project	(The project of building and maintaining IMES was not mentioned.)
Implementation	• The system provides management information on project implementation, so that structured management decisions can be taken. • The following principles are fundamental to the design and implementation of effective interfaces, either for traditional GUI environments or the Web.
Reengineering	(Not mentioned)
Requirements	• This case study serves to illustrate an integrated and practical methodology for evaluating advanced information database systems. The goal of the integration is to create a top-down evaluation process that reduces user and data requirements to a standard evaluation structure. • The team collected and studied the requirements of the system as defined by the main client (European Commission) and set the key evaluation questions. • Efficiency: The degree to which the system realises the planned outputs within the context of the requirements set by the client.
Solution	(Not mentioned)
Point of reference	(Largely technical: The article evaluated IMES based on its inherent quality rather than on how well it was actually used or what difference it made in the work the users were doing or the results of that work.)

Table A.8: Statements providing evidence for an IT perspective vs. business perspective in Vol. 2, Article 17 “Software project Management: A Manager’s View, ” (Jurison, 1999)	
Topic	*Statements from the article*
System	• [This tutorial presents project management] principles and show[s] how they can be applied to the development of information systems. • The planning cycle starts with firming up the goals and objectives and determining the requirements for the system. • In many cases it may be necessary to build and test a prototype to develop a good understanding of the system’s needs and requirements. • Conducted at both unit and system level, [technical reviews] are used to verify the functionality and quality of the system.
User	• Users are often unsure of their needs and frequently change requirements midway through the project. As a result, the software industry is plagued by cost overruns, late deliveries, poor reliability, and user dissatisfaction • Many programmers are introverts and thinking persons who base their decision on facts rather than on feelings and personal values. They often find it difficult to build relationships and see the project from the user's point of view. • User involvement, particularly during the planning phase, leads to better and more realistic definition of system requirements and user commitment to the project
Stakeholder	• Client interaction is particularly important for information systems (IS) projects. As an increasing number of new IS projects become more strategic and involve business process reengineering, management of organizational change is an integral part of project management. • A project can be considered a success only if the client, whether it is a group of internal users or a client in another company, is satisfied with the results. • A clearly defined requirements specification, agreed upon by both the client and the development team, ensures that the client’s needs are correctly understood before starting design work.
IS project	• Projects have specific objectives. Projects must be completed within a specific time period. They have well defined beginnings and ends. Projects must be completed within a given budget. Although some projects may have loosely defined budgets, all projects have budgetary constraints. • The fundamental objective of project management is to "get the job done," to reach the objectives within time, cost, and performance. More recently, managers added a fourth constraint: good client relations. • In information systems projects, performance is specified in terms of certain functional and quality requirements, some of which are quantitative, some qualitative. • All projects can be broadly broken into four generic phases: project conception, planning, execution, termination. The fundamental purpose of the conceptual phase is to determine the feasibility of the project. In the planning phase (sometimes referred to as the definition phase) the performance, cost, and schedule estimates are refined to a point where detailed plans for project execution can be made. In the planning phase

	(sometimes referred to as the definition phase) the performance, cost, and schedule estimates are refined to a point where detailed plans for project execution can be made. • For IS projects, the execution phase frequently extends beyond delivery of the end product and includes system implementation, the process of putting the system into operation in the client's organization. It is not uncommon to have system implementation handled by a separate project team because the implementation team often must function as a change agent rather than as a developer. • A clearly defined requirements specification, agreed upon by both the client and the development team, ensures that the client's needs are correctly understood before starting design work. The requirements document is, in effect, a contract between the client and the development team. • Change control/management, the process of controlling and monitoring changes, is a challenge for all complex projects, but is particularly severe in information systems projects
Implementation	• For IS projects, the execution phase frequently extends beyond delivery of the end product and includes system implementation, the process of putting the system into operation in the client's organization. It is not uncommon to have system implementation handled by a separate project team because the implementation team often must function as a change agent rather than as a developer. • Increasingly IS project managers find themselves playing a central role in their organizations, whether it is an enterprise resource planning (ERP) system implementation, Year 2000 conversion, or a leading-edge technology project.
Reengineering	• As an increasing number of new IS projects become more strategic and involve business process reengineering, management of organizational change is an integral part of project management. • Software requirements are under constant pressure for change. Because software can be changed more easily than hardware, change is a way of life in software development.
Requirements	• Users are often unsure of their needs and frequently change requirements midway through the project. As a result, the software industry is plagued by cost overruns, late deliveries, poor reliability, and user dissatisfaction. • In information systems projects, performance is specified in terms of certain functional and quality requirements, some of which are quantitative, some qualitative. • Project goals, system requirements, project plans, project risks, individual responsibilities, and project status must be visible and understood by all parties involved. • The planning cycle starts with firming up the goals and objectives and determining the requirements for the system. … Clear and unambiguous definition of all deliverables is essential. Technical requirements should be defined early. In many cases it may be necessary to build and test a prototype to develop a good understanding of the system's needs and requirements. A prototype is particularly useful in situations where the client is unsure about the requirements. • [Walkthroughs and inspections] are effective for early detection of errors in requirements, interface prototypes, design, code, and documentation.

	• A clearly defined requirements specification, agreed upon by both the client and the development team, ensures that the client's needs are correctly understood before starting design work. The requirements document is, in effect, a contract between the client and the development team. It specifies what the product must do, but not how. • Even the best prepared requirements specifications will require changes as the software is being designed and tested. Many projects fall victim to "scope creep" caused by uncontrolled changes made well beyond the requirements definition phase. • User involvement, particularly during the planning phase, leads to better and more realistic definition of system requirements and user commitment to the project.
Solution	• [Project meetings] should be attended by appropriate representatives from each major area who can adequately answer questions, negotiate solutions, and make commitments. • Project management packages range from simple schedulers to enterprise-wide solutions and vary in price from about $50 to several thousand dollars.
Point of reference	• [This tutorial presents project management] principles and show[s] how they can be applied to the development of information systems. • Although some projects fail for technical reasons, most project failures are caused by people who ignore the principles of good project management.

Table A.9: Statements providing evidence for an IT perspective vs. business perspective in Vol 2, Article 20, "Dimensions of Information System Success (Seddon, Patnayakuni, Bowtel, 1999)

Topic	*Statements from the article*
System	• [For purposes of evaluating IS success, the authors] define a second dimension, which [they] call System, that is used to classify the type of system that is being evaluated. This dimension has the following six components: - an *aspect* of IT use (e.g., a single algorithm or form of user interface) - a *single* IT application (e.g., a spreadsheet, a PC, or a library cataloging system) - a *type* of IT or IT application (e.g., TCP/IP, a GDSS, a TPS, a data warehouse, etc.) - *all* IT applications used by an organization or sub-organization - an aspect of a system development *methodology* - the *IT function* of an organization or sub-organization.
User	• Grover et al. [1996, p.182] list four different classes of evaluation perspective: (1) users, (2) top management, (3) IS personnel, and (4) external entities. • The IT executive from a local government authority approached the first author of this paper concerned that in a recent survey his IT organization had been criticized as being unresponsive to user needs. • [The article mentions a number of IS evaluation criteria in previous articles: - User acceptance of Expert System advice for expert systems with explanation facilities - Self-rated job performance of users of up to five systems in 25 departments - User Satisfaction as consequence of User participation and four moderator variables

	• For example, the IS effectiveness measures appropriate for evaluating the benefits to an individual user of some aspect of a system might be increased speed of task completion and/or increased decision quality. • Pitt et al. collected opinions from some hundreds of individual users in each firm, so the stakeholders in their study were classified as individual users.
Stakeholder	• Five different types of stakeholders might be considered when evaluating IS success: - The independent observer who is not involved as a stakeholder. - The individual who wants to be better off - The group, which also wants to be better off - The managers or owners who want the organization to be better off - The country which wants the society as a whole to be better off
IS project	(The article does not discuss projects.)
Implementation	(Mentioned in 8 references but not used in the paper.)
Reengineering	(The article does not discuss reengineering.)
Requirements	(This article does not use the term requirement.)
Solution	(The article does not use the term solution.)
Point of reference	• Not surprisingly, a large number of IS effectiveness measures can be found in the IS literature. What is not clear in the literature is what measures are appropriate in a particular context. In this paper we propose a two-dimensional matrix for classifying IS Effectiveness measures.

Table A.10: Statements providing evidence for an IT perspective vs. business perspective in Vol. 2, Article 24, . "Legacy Information Systems and Business Process Change: A Patterns Perspective." (Lloyd, Dewar, and Pooley, 1999)

Topic	*Statements from the article*
System	• The design of large IT systems is extremely hard to separate from the design of business processes. The question then arises: can legacy computer systems 'lock-in' inefficient or even redundant 'legacy' business processes? • This integration-introduction-integration cycle increases the coupling between individual systems that are operated by people for whom many of the couplings/dependencies are hidden within the system. This cycle forms an organisational 'intra-structure' that is typically understood by few people within the organisation (a situation often exacerbated by rounds of downsizing and outsourcing) and becomes a constraint to system redesign that promotes incremental approaches to systems reengineering.

	• Reengineering a system solely to 'improve' its speed … need not bring any competitive advantage …. if the original system was already the fastest part of the overall business process, or…. • Legacy systems can be defined as those that significantly resist modification and evolution to meet business requirements, with a consequentially negative impact on competitiveness. • The literature on the learning organisation also supports this argument. … The patterns of behaviour in large organisations are typically 'hard-wired' into the system through organisational structure, incentive schemes, hiring and promotion practice, and notably information systems. • A system architect, however, lies between the Business and Technology strategists and the application programmer. Although they are responsible for designing systems to support the business strategy and will usually be aware of any constraints that the technology strategy imposes, their knowledge of the specific business strategy, and hence the factors of competition, is likely to be less detailed.
User	(Mentions user requirements, user community, user interface, but does not discuss users directly.)
Stakeholder	(Not discussed directly)
IS project	• This approach [building customized information systems] was accepted for large corporate IT projects during the 1980s. • This analysis was followed with notable IT project failures in the early 1990s, such as …. • We also introduce two patterns drawn from our study of the management of reengineering projects which illustrate how patterns might also be used to capture knowledge about the reengineering process itself. • If a new system is developed to replace part of the old one, the developers will be expected to provide ideal functionality. Consequently, it will be impossible to manage expectations and the project will become large and risky. • You are trying to build a long-term partnership with a supplier on whose support the project's delivery and its long-term success depends
Implementation	• This pattern was used during a *Middleware* implementation. Note that in an organisation-specific reengineering pattern catalogue, this section would also contain contact details of managers involved in the cited implementation. • This pattern was used during a *Divide_and Modernise* implementation. Note that in an organisation-specific reengineering pattern catalogue, this section would also contain contact details of managers involved in the implementation cited. • The systems reengineering pattern chosen in turn leads to consideration of management approaches used in previous implementations of that pattern, and to people within the company who have been responsible for managing this process in the past. The dialogue established through these patterns between different domain experts can be used to confirm the validity of the solution in the current context, help

	establish the composition of the implementation team, and even used to select specific target designs for consideration from a design patterns catalogue. • Pattern languages are recognised for their ability to communicate expertise about technical choices and implementation approaches
Reengineering	• As separate systems become integrated, dependencies are established that complicate future reengineering exercises. • Reengineering these legacy systems to improve competitiveness therefore requires both technical expertise in systems engineering and an understanding of what the business process is intended to achieve. • Reengineering a system solely to 'improve' its speed … need not bring any competitive advantage …. If the original system was already the fastest part of the overall business process, or…. • This theme of re-use is one of a number of guiding principles for developing systems reengineering patterns. • Two generic types of pattern are included: 'reengineering' patterns that relate system characteristics to business and technical imperatives, and 'managing reengineering' patterns which capture knowledge about the reengineering process itself within the context of the organisation. • (Four reengineering patterns are discussed: divide and modernise, wrapping, middleware, and externalising an internal representation. These patterns involve reengineering software rather than business processes..)
Requirements	• Legacy systems can be defined as those that significantly resist modification and evolution to meet business requirements, with a consequentially negative impact on competitiveness. This working definition is chosen carefully from the many alternatives available, because it recognises that a system that is simply 'old' or inflexible is not necessarily a legacy system if there is no business requirement for change. • Even if a requirements explosion does overtake the final restructuring step, the main aim, that of removing the dependency of the functionality on the obsolete technology, will have been achieved. • The work shop may free resources for meeting acute requirements and help build communication links that support earlier identification and response to emergent requirements.
Solution	• Information technology is only part of the over-all solution, and whilst IT is a central enabler of organisational change it is ultimately the business process that constrains the organisation's performance as a whole. • The rise of pre-packaged solutions to common business processes such as accounting and invoicing, produced with economies of scale and benefiting from

	compliance with complex legislation, changed the equation. Smaller companies now found an economic incentive to 'fit' their business process to the standard solution. Large companies …also saw standard solutions provided by market leaders as a means of benchmarking best practice. • During the 1990s, pre-packaged solutions were increasingly accepted by large companies. These companies also saw standard solutions provided by market leaders as a means of benchmarking best practice. • Whilst it can help a company to make its cost base competitive, the values that differentiate its products from the competition may arise from unique aspects of its business process. In these cases, the need to modify a standard solution heavily rather than the business process may eliminate the economic advantage • Alexander identified successful solutions to recurring problems in context, and found a way of communicating these solutions by standardising the format of each pattern and linking related patterns to form a 'pattern language'. In general, therefore, a pattern must contain a description of the *problem* and the *solution*. • [as one of the elements of the system reengineering pattern, a solution is] a proven resolution to a problem. • Hence systems reengineering patterns at, say, a senior management level are likely to capture solutions reflecting a much broader range of concerns than those of a design engineer. • Problem: How can the system interface be made more efficient? Solution: Design an improved user interface and the wrapper shell. The new interface can then invoke the wrapper's API. • Wrapping is often the simplest solution and renders the unsuitable interface invisible to outside users and systems. • A middleware solution was used to integrate the new product offerings with the legacy system, eliminate redundant business processes and to access new functionality,
Point of reference	[Reengineering patterns may be] a means of codifying and disseminating systems reengineering expertise. Through widening the definition of a legacy system to include the business process, we propose that patterns may provide a communication link between business and technology strategists that would help align their objectives and improve the sustainability of any resulting competitive advantage.

APPENDIX III: STATEMENTS FROM *CAIS* ARTICLES, ORGANIZED BY TOPIC

BU	2(04)	System	• A Computer Information System (CIS) …faces continual redevelopment to respond to the changing organisational needs. …Change can only be effective if the plans recognise those who have a stake in the process and they are led to see the value in the new structures or systems. • In establishing a definition appropriate for CIS development it is necessary to consider notions of the system boundary and influences from outside a formal organisation. • Development of new or modified systems in the presence of legacy systems is normal for most companies. Today's new system will become the legacy system in the next, inevitable, round of change.
BU	2(05)	System	• The role of information systems in influencing and enabling organisational design is widely acknowledged. Yet limited attention is paid to the theoretical legitimacy and conceptual basis of IS-enabled organisational change • Most modern change management approaches differentiate from their older counterparts by their focus on the business process as the fundamental unit of analysis in organisational design. According to the perspective they advocate, organisations should not be analysed in terms of the functions into which they can be decomposed or in terms of the products they produce, but in terms of the key business processes that they perform. • Contemporary IS are increasingly integrated together, making it even more difficult to disentangle a single system for evaluation. This may render the demarcation of boundaries around individual systems for the purposes of evaluation a meaningless exercise
BU	2(06)	System	• While information systems continue to serve traditional business needs such as co-ordination of production and enhancements of services offered, a new and important role has emerged: the potential for such systems to adopt a supervisory and strategic support role. • Any type of change whether it involves the development of a computerised system or the re-engineering of business processes involves many assumptions about the embedding enterprise domain. • Prior to designing new business processes and support information systems, any reform requires a clear understanding (and a sharing of this understanding among many stakeholders) of the current enterprise situation.
BU	2(07)	System	• Legacy information systems can represent a huge investment for organizations in terms of information technology, business processes, procedures and organizational structures. • Legacy information systems include business and technical dimensions … the systems can present problems when there is a misalignment between the strategic vision of the business, the IT legacy and the old business model embodied in the legacy.

			• Legacy information systems are defined as information technology (e.g. hardware, software applications and network) and the business model implicit in the application of that technology (e.g. organizational structure, work flows, procedures and processes) within the organization. • First-generation systems dating from the 1960s and 1970s were in machine language but most were developed in assembly or early versions of third-generation programming languages such as COBOL or FORTRAN • Second-generation systems (late 1970s and throughout the 1980s) possessed some degree of modularity and many were used for online transaction processing. •The real value to the organization of information technology legacy systems lies in the "accumulation of years of business rules, policies, expertise and 'knowhow' embedded in the system. • The business legacy is embedded in the legacy IT system, and it is the inter-relatedness of business and IT legacy which makes either business or technical change a difficult process. • Interdependence can make it difficult to predict changes arising in the system as a whole as a result of minor enhancements to one component.
BU	2(08)	System	• 'Legacy', we argue, is not just a problem encountered by organisations with aging mainframes and dated software, it is an issue from the moment a computer system becomes an integral part of any organisation's everyday work. • 'The system as a whole, and both main software packages, were seen as 'dated', 'slow', prone to unpredictable breakdown, and not 'user friendly'. • The main 'workhorse' systems in use in the bank were two software packages: BAF, an accounting/bookkeeping package dating from the 1960s, that, "had bits bolted onto it"; and ISS, a more modern relational database • Financial institutions were among the first wave of business organisations to computerise many of their operations. A great deal of their basic functioning is now dependent on those aging systems. • You've got to use it as a tool…using the software to confirm rather than determine decisions … may have arisen as a consequence of the inclusion in the program of 'non-financial' information which could significantly influence the risk grade obtained. • Workers were required to indicate that they had completed all the formalities on each screen before they would be permitted (by the machine) to proceed to the next. This rigid workflow model would, however, occasionally create problems… there were occasions when they needed to subvert the strict workflow model.
BU	2(20)	System	• [For purposes of evaluating IS success, the authors] define a second dimension, which [they] call System, that is used to classify the type of

			system that is being evaluated. This dimension has the following six components: - an *aspect* of IT use (e.g., a single algorithm or form of user interface) - a *single* IT application (e.g., a spreadsheet, a PC, or a library cataloging system) - a *type* of IT or IT application (e.g., TCP/IP, a GDSS, a TPS, a data warehouse, etc.) - *all* IT applications used by an organization or sub-organization - an aspect of a system development *methodology* - the *IT function* of an organization or sub-organization.
IT	2(03)	System	• This paper presents a successful new approach which focuses primarily on the architecture of the software system to migrate an existing system to a new form. • Prioritization of requirements is typically dictated, *in the final analysis*, by the business context the software system serves, but in the first analysis there are usually a host of hidden assumptions underlying the business perspective itself. • Expertise in shifting legacy systems to new paradigms is buried in the folklore of software engineering. • However, the movement of any large-scale business-critical system to components is fraught with difficulty. • Legacy systems have been defined as stand-alone applications built during a prior era's technology but they are perhaps more widely understood as software systems whose plans and documentation are either poor or non-existent.. A more useful definition … is: "A legacy system is a large system delivering significant business value today from a substantial pre-investment in hardware and software that may be many years old. … It is a business-critical system which has an architecture which makes it insufficiently flexible to meet the challenges of anticipated future change requirements" • Meanwhile successful systems have simply aged, some less gracefully than others. Jones estimates that the average rate of change of software systems is between 5% and 7% every year, year on year. • Mergers, takeovers, shutdowns and corporate restructuring can turn well-planned, well-engineered up-to-date systems into obsolescence virtually overnight. • "How do we best support the sale of new financial products, and what is the optimum configuration of IT for this purpose?". The solution may indeed involve new software development to replace the old system, or it may mean simple, incremental enhancement of the old system, or a mixture of both. The point is that the legacy problem is primarily a business problem, and only incidentally a technical one, and that any solution must be driven from the problem space.
IT	2(15)	System	• IMES is an integrated information system that incorporates Internet technologies to provide wide monitoring and evaluation capabilities. It consists of five individual, but interacting, subsystems that form a robust intranet information system. The subsystems are the database

			(thousands of management reports from 1994 to 1999), the local application, the input/output assistants, the intranet component, and the security mechanism. • The system provides management information on project implementation, so that structured management decisions can be taken. • The VB Script language, which creates these Web pages, submits calls to the system database using ODBC driver technologies. • This database is the "back end" application used for storing all kinds of data. It is built in MS Access 7.0 and structured according to the relational model into entities and relationships.
IT	2(17)	System	• [This tutorial presents project management] principles and show[s] how they can be applied to the development of information systems. • The planning cycle starts with firming up the goals and objectives and determining the requirements for the system. • In many cases it may be necessary to build and test a prototype to develop a good understanding of the system's needs and requirements. • Conducted at both unit and system level, [technical reviews] are used to verify the functionality and quality of the system.
IT	2(24)	System	• The design of large IT systems is extremely hard to separate from the design of business processes. The question then arises: can legacy computer systems 'lock-in' inefficient or even redundant 'legacy' business processes? • This integration-introduction-integration cycle increases the coupling between individual systems that are operated by people for whom many of the couplings/dependencies are hidden within the system. This cycle forms an organisational 'intra-structure' that is typically understood by few people within the organisation (a situation often exacerbated by rounds of downsizing and outsourcing) and becomes a constraint to system redesign that promotes incremental approaches to systems reengineering. • Reengineering a system solely to 'improve' its speed … need not bring any competitive advantage …. if the original system was already the fastest part of the overall business process, or…. • Legacy systems can be defined as those that significantly resist modification and evolution to meet business requirements, with a consequentially negative impact on competitiveness. • The literature on the learning organisation also supports this argument. … The patterns of behaviour in large organisations are typically 'hard-wired' into the system through organisational structure, incentive schemes, hiring and promotion practice, and notably information systems. • A system architect, however, lies between the Business and Technology strategists and the application programmer. Although they are responsible for designing systems to support the business strategy

			and will usually be aware of any constraints that the technology strategy imposes, their knowledge of the specific business strategy, and hence the factors of competition, is likely to be less detailed.

BU	2(04)	User	• Computer Information Systems development often focuses on direct users and affected internal departments as the exclusive stakeholders. However these groups may present too narrow a perspective. • Looking beyond the immediate users of the system greatly increases the number of views which need to be addressed.
BU	2(05)	User	• Many systems still fail to fulfil the needs of their users and the organisations that adopt them.
BU	2(06)	User	• [mentions users of the EKD framework, as in:] -The EKD roadmap is a navigational structure in the sense that it allows the roadmap user to determine their route between the different knowledge states regarding organisational change. - For example, if the user has no knowledge about the organisation then the entry point will be the Null state. - Thus, the electronic roadmap can be used by different users at different sites in the organisation.
BU	2(07)	User	(Not used in significant way in article.)
BU	2(08)	User	• Computer systems have been installed in many companies for some time now and no matter how well they may have fitted the situation initially, usage and the circumstances of use have changed, as indeed have the needs and the users, and, most importantly, the organisations themselves • Observation and conversations with users indicated a number of problems. The system as a whole, and both main software packages, were seen as ‘dated’, ‘slow’, prone to unpredictable breakdown, and not ‘user friendly’.
BU	2(20)	User	• Grover et al. [1996, p.182] list four different classes of evaluation perspective: (1) users, (2) top management, (3) IS personnel, and (4) external entities. • The IT executive from a local government authority approached the first author of this paper concerned that in a recent survey his IT organization had been criticized as being unresponsive to user needs. • [The article mentions a number of IS evaluation criteria in previous articles: - User acceptance of Expert System advice for expert systems with explanation facilities

			- Self-rated job performance of users of up to five systems in 25 departments - User Satisfaction as consequence of User participation and four moderator variables • For example, the IS effectiveness measures appropriate for evaluating the benefits to an individual user of some aspect of a system might be increased speed of task completion and/or increased decision quality. • Pitt et al. collected opinions from some hundreds of individual users in each firm, so the stakeholders in their study were classified as individual users.
IT	2(03)	User	• …The development team was able to capture the essence of the business problem by working with the users. • One of Alexander's first contributions was to reject the modern split between architect (who theorizes) and builder (who constructs, following the architect's drawings) in favour of a combination of user (inhabitant)-centred design and an architect-builder model in which the architect also implements
IT	2(15)	User	• To most users, the interface is the system. • Effective interfaces do not concern the user with the inner workings of the system • [The information system] [provides relevant data to the users] and provides [search capabilities]. • [Evaluation criteria include:] - Reliability: The extent to which the clients can trust the system and its services - Accessibility: The degree to which the system database is easy to be accessed by the users. - Ease of use: The extent to which the users can "navigate" in the system database and use its services.
IT	2(17)	User	• Users are often unsure of their needs and frequently change requirements midway through the project. As a result, the software industry is plagued by cost overruns, late deliveries, poor reliability, and user dissatisfaction • Many programmers are introverts and thinking persons who base their decision on facts rather than on feelings and personal values. They often find it difficult to build relationships and see the project from the user's point of view. • User involvement, particularly during the planning phase, leads to better and more realistic definition of system requirements and user commitment to the project
IT	2(24)	User	(Mentions user requirements, user community, user interface, but does not discuss users directly.)

BU	2(04)	Stakeholder	• For our purposes a stakeholder is someone who has an interest in a CIS development *and* can affect the success of that development. • Management of change, at all levels, needs to be informed and endorsed. Change can only be effective if the plans recognise those who have a stake in the process and they are led to see the value in the new structures or systems. • A particular risk is that close to the technical boundary we will find stakeholders who have extreme views of existing, or legacy systems. The danger is that inappropriate factors may be given more weight than the wider needs of the organisation and its environment. Stakeholders close to the technology can be expected to express their personal investment in the current technology, their detailed experience of operational problems, or their technological bias lending enthusiasm for the promises of new technology.
BU	2(05)	Stakeholder	• Project champions tend to underestimate costs and overestimate benefits. • Interviews with key process participants (management and employees) of both companies were conducted to capture the process essence and decompose the order fulfillment process into its component activities. The knowledge elicited by the interviews was used to define the boundaries of the process and the models to be developed.
BU	2(06)	Stakeholder	• The implications of these forces on this organization [part of an electricity company] is that, prior to designing new business processes and support information systems, any reform requires a clear understanding (and a sharing of this understanding among many stakeholders). • Both scenario evaluation as well as interpretation of evaluation data was dependent on subjective judgement of involved participants. Finally, it should be noted that the evaluation data provided organisational stakeholders with a rationale means of making an informed choice.
BU	2(07)	Stakeholder	• The inclination and acceptance of change does not exist within the culture of the organization and hence employees resist change. Although managers see change as an opportunity to strengthen the business, employees may perceive change as disruptive and intrusive.
BU	2(08)	Stakeholder	(Not mentioned explicitly. Mentioned implicitly in statements about users and about IS projects)
BU	2(20)	Stakeholder	• Five different types of stakeholders might be considered when evaluating IS success: - The independent observer who is not involved as a stakeholder. - The individual who wants to be better off - The group, which also wants to be better off - The managers or owners who want the organization to be better off

			- The country which wants the society as a whole to be better off
IT	2(03)	Stakeholder	(This term is not used, but the term user seems to be a synonym in this paper.)
IT	2(15)	Stakeholder	(Stakeholders other than direct users were not mentioned directly.)
IT	2(17)	Stakeholder	• Client interaction is particularly important for information systems (IS) projects. As an increasing number of new IS projects become more strategic and involve business process reengineering, management of organizational change is an integral part of project management. • A project can be considered a success only if the client, whether it is a group of internal users or a client in another company, is satisfied with the results. • A clearly defined requirements specification, agreed upon by both the client and the development team, ensures that the client's needs are correctly understood before starting design work.
IT	2(24)	Stakeholder	(Not discussed directly)

BU	2(04)	IS project	• Computer Information Systems development often focuses on direct users and affected internal departments as the exclusive stakeholders. However these groups may present too narrow a perspective. To improve the effectiveness of the development process, a wider constituency should be considered that includes organisational partners in the wider business environment. • [The following themes] capture a pragmatic dimension to justifying a "stakeholder's" participation. - Stakeholders may affect realisation or may be affected by realisation of a system; - Stakeholders may have actual versus legitimate influence; they may be an internal affect or versus external; - Stakeholders may have a supportive influence versus conflictive influence - They may be stakeholders of a common value; they therefore need to be considered, consulted, participative, or responsible for process under consideration or development by the system.
BU	2(05)	IS project	• SDLC-based IS development methods …perpetuate the distinction between the business and the IS domain. Most structured approaches to IS development begin with an implicit assumption that the business domain issues are resolved and the system is to work in a stable and well-defined business environment, where the only issue is to identify the 'correct' requirements for the new IS. As a result, not enough attention is generally being paid to investigating the interactions of the IS to be developed with the business processes it will naturally affect.

			• The design and implementation of information systems is generally a complex and laborious exercise for most contemporary organisations. It may not be desirable (or even feasible) to incorporate such design into business process change in its entirety. A strategy where IS design is treated along two dimensions (one concerning the organisational impact of IS, and the other concerning the technical implementation details) may be more appropriate.
BU	2(06)	IS project	• Change goals form the requirements upon which the re-engineered enterprise structure will be based. This task concerns the mapping of change requirements onto a future enterprise model, which in turn involves the modelling of the future enterprise goals and how these goals will be realised in terms of operational enterprise components. • The EKD framework for modelling organisational change defines the set of applicable knowledge states that need to be reached in an organisational change project. However, it does not dictate any particular ordering between these states…. Instead, each state to be reached is dynamically selected in the course of the change management process. Each route characterises a specific method for solving the problem at hand.
BU	2(07)	IS project	• Legacy systems comprise the existing characteristics of an organization such as its structures, processes, strategy and cultures resulting from the impact of internal and external forces. These characteristics can mean that even when new emerging technologies are introduced into organizations, employees have difficulty in adapting to new ways of working.
BU	2(08)	IS project	• It is unlikely that any organisation is ever 'going to get it right' the first time. What it *does* suggest is the need for more effective monitoring of new technologies in their situations of use and developing effective mechanisms for involving users' experiences in development. • No matter how promising new technologies may seem, the realities of their implementation are typically disruptive. They also involve huge overheads in respect of retraining and compensatory payments, not to mention the lead-times required for familiarisation of workers using the system. These problems arise whenever new systems are introduced, no matter how carefully the planning was done. • Part of the 'centralising' objective was an attempt to ensure that, for every single process in which the bank engaged, there would be a process map so that anyone could come in and do the job in exactly the same way as anyone else. … it was necessary for workers involved in different aspects of the lending process to arrive at some sort of understanding of the work of others involved in the same process, beyond their own teams, and sometimes beyond the walls of the Lending Centre itself.
BU	2(20)	IS project	(The article does not discuss projects.)
IT	2(03)	IS project	• The *raison d'être* for contemplating a move to an object-based representation for an existing system is the belief that business benefits in terms of increased flexibility to business change, and increased productivity (through software reuse) will result. •All the projects were considered to be successful in terms of their immediate technical objectives, their medium to long-term business objectives, and in their strategic and tactical research objectives.

			• Legacy information systems are typically the targets of reverse engineering projects. • Best practice is captured in the form of software patterns that address not only the design, but crucially also the process and organizational issues that inevitably surround such a project. • The business case [is] the key criteria for determining whether or not to migrate a legacy system and …the software architecture [is] the main focus of attention for the migration process.
IT	2(15)	IS project	(The project of building and maintaining IMES was not mentioned.) • The goals of IMES included: - Improve management reporting on Tacis progress and results. - Improve the management of the monitoring contracts. - Future planning of further …activities
IT	2(17)	IS project	• Projects have specific objectives. Projects must be completed within a specific time period. They have well defined beginnings and ends. Projects must be completed within a given budget. Although some projects may have loosely defined budgets, all projects have budgetary constraints. • The fundamental objective of project management is to "get the job done," to reach the objectives within time, cost, and performance. More recently, managers added a fourth constraint: good client relations. • In information systems projects, performance is specified in terms of certain functional and quality requirements, some of which are quantitative, some qualitative. • All projects can be broadly broken into four generic phases: project conception, planning, execution, termination. The fundamental purpose of the conceptual phase is to determine the feasibility of the project. In the planning phase (sometimes referred to as the definition phase) the performance, cost, and schedule estimates are refined to a point where detailed plans for project execution can be made. In the planning phase (sometimes referred to as the definition phase) the performance, cost, and schedule estimates are refined to a point where detailed plans for project execution can be made. • For IS projects, the execution phase frequently extends beyond delivery of the end product and includes system implementation, the process of putting the system into operation in the client's organization. It is not uncommon to have system implementation handled by a separate project team because the implementation team often must function as a change agent rather than as a developer. • A clearly defined requirements specification, agreed upon by both the client and the development team, ensures that the client's needs are correctly understood before starting design work. The requirements document is, in effect, a contract between the client and the development team.

			• Change control/management, the process of controlling and monitoring changes, is a challenge for all complex projects, but is particularly severe in information systems projects
IT	2(24)	IS project	• This approach [building customized information systems] was accepted for large corporate IT projects during the 1980s. • This analysis was followed with notable IT project failures in the early 1990s, such as …. • We also introduce two patterns drawn from our study of the management of reengineering projects which illustrate how patterns might also be used to capture knowledge about the reengineering process itself. • If a new system is developed to replace part of the old one, the developers will be expected to provide ideal functionality. Consequently, it will be impossible to manage expectations and the project will become large and risky. • You are trying to build a long-term partnership with a supplier on whose support the project's delivery and its long-term success depends

BU	2(04)	Implementation	• Another failing in the group's decision making was the difficulty in separating technology and implementation from strategic decision making.
BU	2(05)	Implementation	• This point reinforces our earlier argument for incorporating only the high-level organisational impacts of IS in business process design and leaving the low-level technical implementation details for later. • Finally, SDLC-based approaches tend to view IS evaluation as a post-implementation activity, addressed only in the last step of the system development life cycle. • What may be needed is an explicit focus on the pre-implementation (*ex ante*) evaluation of the information system (for example, within the problem identification or system analysis stages). • [IS] evaluation provides the benchmarks of what is to be achieved by the IS investment. These benchmarks can later be used to provide a measure of the actual implementation success of IS projects. • Further to the simulation analysis, the process scenarios were scrutinised to develop a detailed understanding of implementation

			challenges and transform hypotheses into detailed implementation plans. The requirements of each option regarding technology, people, and skills were assessed and a formal cost-benefit analysis was conducted to evaluate the proposed investments. • The challenge for business engineering is to bring process design and IS design together without adding to the high complexity of each task alone. … A potential strategy for addressing this need would involve incorporating high-level IS design and IS evaluation into business process design, and leaving the technical details of IS implementation to be addressed in the aftermath of business engineering decisions. Such an approach was followed in the case study where the EDI applications were defined in general terms (only to the level of detail necessary for the model development and analysis) without the need for specific reference to implementation-dependent technical details. What is even more important is that implementation details need only be developed for the solution chosen and not for every alternative information system design that was considered during the business engineering endeavour. • The design and implementation of information systems is generally a complex and laborious exercise for most contemporary organisations. …A strategy where IS design is treated along two dimensions (one concerning the organisational impact of IS, and the other concerning the technical implementation details) may be more appropriate. • Such methodologies should satisfy the requirements identified above, namely adopting a process perspective in analysing organisational structures, integrating high-level IS design within business process design, and leaving the technical details of IS implementation to the software engineering domain experts.
BU	2(06)	Implementation	• The implementation comprises introducing customers profiling, minimising delay time to serve an application, offering all means for payment, offering all services at customer premises, introducing all available technologies to communicate with customers, introducing IT solutions for all services.
BU	2(07)	Implementation	• The existence of legacy systems that hinder the implementation of new business strategies is now well established. • The company decided to reengineer the organization including implementation of a process-oriented ERP solution.
BU	2(08)	Implementation	• No matter how promising new technologies may seem, the realities of their implementation are typically disruptive.
BU	2(20)	Implementation	(Mentioned in 8 references but not used in the paper.)
IT	2(03)	Implementation	• When a shift is being contemplated from, say, representation in a structured language to representation in an object-oriented implementation, it is not just the language that is changing but the development paradigm itself. • The architecture is did *not* necessarily imply an object-oriented *implementation*. Indeed, the first two pioneering projects delivered a

			restructured system in the same base technology in which the legacy system was originally implemented. • Irrespective of the target implementation technology, object modelling was used to capture a description of the existing system in its business context, describe the new architecture, and plan the technical migration. • The next steps largely concern the detail of the implementation abstractions, which will, of course, include legacy code. • This scoping of the analysis model so that it captured the key abstractions of the problem space and modelled them separately and independently of any implementation concerns reflects the *Shamrock* pattern of the ADAPTOR language. • By postponing consideration of the *representation* of the key VAT abstractions in software, and of their implementation and interfacing to other components in the customer service system, the development team was able to capture the essence of the business problem by working with the users. • The basic notion is that classes which exist to access legacy code should differ from other objects only in their implementation details.
IT	2(15)	Implementation	• The system provides management information on project implementation, so that structured management decisions can be taken. • The following principles are fundamental to the design and implementation of effective interfaces, either for traditional GUI environments or the Web.
IT	2(17)	Implementation	• For IS projects, the execution phase frequently extends beyond delivery of the end product and includes system implementation, the process of putting the system into operation in the client's organization. It is not uncommon to have system implementation handled by a separate project team because the implementation team often must function as a change agent rather than as a developer. • Increasingly IS project managers find themselves playing a central role in their organizations, whether it is an enterprise resource planning (ERP) system implementation, Year 2000 conversion, or a leading-edge technology project.
IT	2(24)	Implementation	• This pattern was used during a *Middleware* implementation. Note that in an organisation-specific reengineering pattern catalogue, this section would also contain contact details of managers involved in the cited implementation. • This pattern was used during a *Divide_and Modernise* implementation. Note that in an organisation-specific reengineering pattern catalogue, this section would also contain contact details of managers involved in the implementation cited. • The systems reengineering pattern chosen in turn leads to consideration of management approaches used in previous implementations of that pattern, and to people within the company who have been responsible for

			managing this process in the past. The dialogue established through these patterns between different domain experts can be used to confirm the validity of the solution in the current context, help establish the composition of the implementation team, and even used to select specific target designs for consideration from a design patterns catalogue. • Pattern languages are recognised for their ability to communicate expertise about technical choices and implementation approaches

BU	2(04)	Reengineering	(Not discussed in the article)
BU	2(05)	Reengineering	• Business engineering is defined here as the integral, concurrent design of organisational processes and the information systems to support them. • The challenge for business engineering is to bring process design and IS design together without adding to the already high complexity of each task alone. One way to achieve unity is to incorporate high-level IS design into business process design projects and leave the technical details of IS implementation to be managed in the aftermath of process change decisions.
BU	2(06)	Reengineering	• Any type of change whether it involves the development of a computerised system or the re-engineering of business processes involves many assumptions about the embedding enterprise domain. • In a business process re-engineering project, one may start by understanding the current situation (reach the As-Is state) and proceed with exploring alternative change scenarios (reach the Change state), continuing with the evaluation of alternative scenarios (reach the Evaluation state) and finally, design the re-engineered business processes according to the selected change plan (reach the To-Be state).
BU	2(07)	Reengineering	• Many organizations are finding that legacy information systems act as a barrier to strategic innovation. • The gap between what the legacy information systems can deliver and the strategic vision of the organization widens when the legacy information systems are unable to adapt to meet the new requirements.
BU	2(08)	Reengineering	(The term reengineering is not used, but this seems to capture the authors' view of reengineering:.) • "…. Apparently small changes may have major implications." This statement is not necessarily an indication of the unwillingness of those responsible for the development of the system to make appropriate changes. It is equally likely to be an indication of just how difficult it is to modify systems which are already in use and upon which the work depends, not to mention the problems of technical complexity. In significant respects, problems such as these are as much organisational as technological because they direct attention to the need to reorganise work and implement new technologies in a more integrated way.

BU	2(20)	Reengineering	(The article does not discuss reengineering.)
IT	2(03)	Reengineering	• The migration of legacy systems is a process of re-engineering. The accepted definition of re-engineering is … "the examination and alteration of the target system to reconstitute it in a new form". • There is a particular quality to the re-engineering effort that must be understood when it involves moving a computer system from, say, a structured representation to an object-based one, however. • The relative failure of traditional reverse engineering techniques when applied to the restructuring of systems to an object-based or object-oriented form results from their tendency to ignore the changing problem space which, typically, is driving the need for change in the first place.
IT	2(15)	Reengineering	(Not mentioned)
IT	2(17)	Reengineering	• As an increasing number of new IS projects become more strategic and involve business process reengineering, management of organizational change is an integral part of project management. • Software requirements are under constant pressure for change. Because software can be changed more easily than hardware, change is a way of life in software development.
IT	2(24)	Reengineering	• As separate systems become integrated, dependencies are established that complicate future reengineering exercises. • Reengineering these legacy systems to improve competitiveness therefore requires both technical expertise in systems engineering and an understanding of what the business process is intended to achieve. • Reengineering a system solely to 'improve' its speed … need not bring any competitive advantage …. If the original system was already the fastest part of the overall business process, or…. • This theme of re-use is one of a number of guiding principles for developing systems reengineering patterns. • Two generic types of pattern are included: ‘reengineering’ patterns that relate system characteristics to business and technical imperatives, and ‘managing reengineering’ patterns which capture knowledge about the reengineering process itself within the context of the organisation. • (Four reengineering patterns are discussed: divide and modernise, wrapping, middleware, and externalising an internal representation. These patterns involve reengineering software rather than business processes..)

BU	2(04)	Requirements	• [In relation to a case study] Many of these external stakeholders are governmental bodies whose needs for the supply of data and reports are an integral requirement of any university's student record keeping package. • [Paul] identifies six environmental changes that have the potential to affect an organisation's CIS needs. [The first is] changes in legal requirements.
BU	2(05)	Requirements	• Most structured approaches to IS development begin with an implicit assumption that the business domain issues are resolved and the system is to work in a stable and well-defined business environment, where the only issue is to identify the 'correct' requirements for the new IS. • The life span of IS is uncertain (due to technological obsolescence and changing requirements). • The requirements of each option regarding technology, people, and skills were assessed and a formal cost-benefit analysis was conducted to evaluate the proposed investments. • ..Although most existing IS development methods begin by stressing the importance of understanding the real-world operation that the IS will support, they quickly become absorbed in the definition of individual functions and detailed requirements ('reductionism').
BU	2(06)	Requirements	• Change goals form the requirements upon which the re-engineered enterprise structure will be based. This task concerns the mapping of change requirements onto a future enterprise model, which in turn involves the modelling of the future enterprise goals and how these goals will be realised in terms of operational enterprise components. • Using the EKD ends-means links, change in enterprise goals (regarding for example, company objectives, policy, general market condition) will propagate top-down as reasons or requirements for re-organising the enterprise processes. • These sessions resulted in the specification of both internal enterprise needs as well as external constraints that defined the enterprise change requirements. • GroupSystems is a suite of team-based decision software tools that were used for the identification, elaboration and resolution of stakeholder requirements. • Having agreed on a set of change requirements the next step in our route was to identify how these requirements could be compared and contrasted with the current goals, thus providing a basis for a reasoned approach for future improvement. This task resulted in the identification of alternative change scenarios indicating the type of organisational transformation necessary for satisfying change requirements.
BU	2(07)	Requirements	• In addition to date and regulation requirements, a range of business pressures are increasingly significant today.

			pressures are increasingly significant today •The information systems were not integrated between sites or within sites. They could not support the MIS requirements for a profit oriented business. i.e. measure profitability and monitor operating costs
BU	2(08)	Requirements	• The two managers had to work together to produce some kind of model that seemed to give due consideration to their own, highly particular requirements. The end product was a complex and highly creative design that was heavily informed by their own experience of the day-to-day character of their work, and the work of the staff around them.
BU	2(20)	Requirements	(This article does not use the term requirement.)
IT	2(03)	Requirements	• [A legacy system] is a business-critical system which has an architecture which makes it insufficiently flexible to meet the challenges of anticipated future change requirements. • A software system that tries to meet the requirements of all possible scenarios will almost certainly suffer 'analysis paralysis' and will be too complicated and/or inefficient to deliver and use. • A software architecture that maps closely onto the key abstractions …[increases] the likelihood of maintaining traceability from solutions to requirements through such business-driven changes. • In each case the systems' owners made a business decision that they needed a component-based architecture in order to meet the challenge of ever new requirements, and this architecture implied the kind of encapsulation that object-based systems deliver. • This understanding frees the developer to utilize the same requirements gathering and modelling techniques to describe any part of a system that could be used to describe the system as a whole. • The utilization of use cases to capture the 'as is' requirements reflects pattern 22, *Scenarios Define Problem* in Coplien's organization and process pattern language.
IT	2(15)	Requirements	• This case study serves to illustrate an integrated and practical methodology for evaluating advanced information database systems. The goal of the integration is to create a top-down evaluation process that reduces user and data requirements to a standard evaluation structure. • The team collected and studied the requirements of the system as defined by the main client (European Commission) and set the key evaluation questions. • Efficiency: The degree to which the system realises the planned outputs within the context of the requirements set by the client.
IT	2(17)	Requirements	• Users are often unsure of their needs and frequently change requirements midway through the project. As a result, the software industry is plagued by cost overruns, late deliveries, poor reliability, and user dissatisfaction.

			• In information systems projects, performance is specified in terms of certain functional and quality requirements, some of which are quantitative, some qualitative. • Project goals, system requirements, project plans, project risks, individual responsibilities, and project status must be visible and understood by all parties involved. • The planning cycle starts with firming up the goals and objectives and determining the requirements for the system. … Clear and unambiguous definition of all deliverables is essential. Technical requirements should be defined early. In many cases it may be necessary to build and test a prototype to develop a good understanding of the system's needs and requirements. A prototype is particularly useful in situations where the client is unsure about the requirements. • [Walkthroughs and inspections] are effective for early detection of errors in requirements, interface prototypes, design, code, and documentation. • A clearly defined requirements specification, agreed upon by both the client and the development team, ensures that the client's needs are correctly understood before starting design work. The requirements document is, in effect, a contract between the client and the development team. It specifies what the product must do, but not how. • Even the best prepared requirements specifications will require changes as the software is being designed and tested. Many projects fall victim to "scope creep" caused by uncontrolled changes made well beyond the requirements definition phase. • User involvement, particularly during the planning phase, leads to better and more realistic definition of system requirements and user commitment to the project.
IT	2(24)	Requirements	• Legacy systems can be defined as those that significantly resist modification and evolution to meet business requirements, with a consequentially negative impact on competitiveness. This working definition is chosen carefully from the many alternatives available, because it recognises that a system that is simply 'old' or inflexible is not necessarily a legacy system if there is no business requirement for change. • Even if a requirements explosion does overtake the final restructuring step, the main aim, that of removing the dependency of the functionality on the obsolete technology, will have been achieved. • The work shop may free resources for meeting acute requirements and help build communication links that support earlier identification and response to emergent requirements.

BU	2(04)	Solution	• Initial planning was at a level of detail that had to be discarded when the emergence of Internet technology and applications provided a readily implementable solution in 1995
BU	2(05)	Solution	• In line with the previous analysis, business process simulation was employed to assist in identifying the problems of existing process designs, to formulate appropriate solutions based on EDI applications, and to realise the expected impacts of these solutions on key business performance indicators. • Simulation made it possible to realise that, if combined with the technology introduction, other (non EDI-dependent) structural process changes could provide a solution to the inefficiencies of the process.
BU	2(06)	Solution	• The aim of evaluation is to deliver an enterprise model, which is consistent with the stakeholders' experience and/or expectations. Often, alternative enterprise models may be possible (e.g., there may be multiple change models, leading to alternative future solutions). • This approach focuses on the systematic analysis of the effects of change requirements on the existing enterprise context, rather than prescribing a solution based on experts' opinions
BU	2(07)	Solution	• Developments in information technology add to the problem [of enhancing existing systems] as technology moves beyond traditional transaction processing towards client/server architectures and the Internet to create new types of business solutions. • The company decided to reengineer the organization including implementation of a process-oriented ERP solution. • …decided to move towards a process-oriented approach facilitated by an ERP solution.
BU	2(08)	Solution	• Ethnographic methods … bring a particular focus to the analysis of systems in use and thereby outline the 'play of possibilities' for work and design, "enabling designers to question the taken-for-granted assumptions embedded in the conventional problem-solution-design framework."
BU	2(20)	Solution	(The article does not use the term solution.)
IT	2(03)	Solution	• The spread of the PC from the mid-1980s encouraged a culture in which 'point solutions' were developed. • Similar applications running on different operating systems on different boxes became common. Worse still, key business abstractions such as 'Customer' could be running on different applications on the same machine at the same time, and since these applications could not talk to each other, information integrity could not be maintained. Subsequently, such point solutions became subject to localized optimizations, and uncontrolled maintenance, etc., exacerbating the position even further • But these benefits rely, as we have seen, on the fact that object systems 'break' from the underlying Von Neuman architecture of the machine and enable the possibility of building software solutions in the image of the problem space itself. • A software architecture that maps closely onto the key abstractions

			…[increases] the likelihood of maintaining traceability from solutions to requirements through such business-driven changes.
IT	2(15)	Solution	(Not mentioned)
IT	2(17)	Solution	• [Project meetings] should be attended by appropriate representatives from each major area who can adequately answer questions, negotiate solutions, and make commitments. • Project management packages range from simple schedulers to enterprise-wide solutions and vary in price from about $50 to several thousand dollars.
IT	2(24)	Solution	• Information technology is only part of the over-all solution, and whilst IT is a central enabler of organisational change it is ultimately the business process that constrains the organisation's performance as a whole. • The rise of pre-packaged solutions to common business processes such as accounting and invoicing, produced with economies of scale and benefiting from compliance with complex legislation, changed the equation. Smaller companies now found an economic incentive to 'fit' their business process to the standard solution. Large companies …also saw standard solutions provided by market leaders as a means of benchmarking best practice. • During the 1990s, pre-packaged solutions were increasingly accepted by large companies. These companies also saw standard solutions provided by market leaders as a means of benchmarking best practice. • Whilst it can help a company to make its cost base competitive, the values that differentiate its products from the competition may arise from unique aspects of its business process. In these cases, the need to modify a standard solution heavily rather than the business process may eliminate the economic advantage • Alexander identified successful solutions to recurring problems in context, and found a way of communicating these solutions by standardising the format of each pattern and linking related patterns to form a 'pattern language'. In general, therefore, a pattern must contain a description of the *problem* and the *solution*. • [as one of the elements of the system reengineering pattern, a solution is] a proven resolution to a problem. • Hence systems reengineering patterns at, say, a senior management level are likely to capture solutions reflecting a much broader range of concerns than those of a design engineer. • Problem: How can the system interface be made more efficient? Solution: Design an improved user interface and the wrapper shell. The new interface can then invoke the wrapper's API. • Wrapping is often the simplest solution and renders the unsuitable interface invisible to outside users and systems.

			• A middleware solution was used to integrate the new product offerings with the legacy system, eliminate redundant business processes and to access new functionality,

BU	2(04)	Point of reference	• This paper presents a method, the stakeholder web that identifies appropriate stakeholders and their viewpoints.
BU	2(05)	Point of reference	• We are aware of no IS evaluation method that actually advocates such a perspective [using the business process as the unit of analysis] for appraising the benefits of an information system by measuring the impact of changes on the level of the business processes that the IS is intended to support. • We need to adopt process change as a mediating factor between the IS initiative and economic return. Such thinking could trigger a radically different perspective in the way IS investments are viewed and analysed within an organisation.
BU	2(06)	Point of reference	• Modelling of organisational change in EKD is achieved through the use of: a common set of concepts for describing enterprise knowledge regarding organisational change, i.e., the EKD enterprise ontology and a methodology roadmap and associated guidelines for assisting user navigation within the space of the possible routes connecting the four knowledge states (As-is, Change, To-Be, and Evaluation).
BU	2(07)	Point of reference	• Legacy information systems are usually considered from a technical perspective, addressing issues such as age, complexity, maintainability, design and technology. We wish to demonstrate that the business dimension to legacy information systems, represented by the organisation structure, business processes and procedures that are bound up in the design and operation of the existing IT systems, is also significant.
BU	2(08)	Point of reference	• Legacy concerns are not merely technological in focus but also organizational in the sense of being intimately wrapped up with the everyday accomplishment of work. • The rapidly changing nature of commercial and organisational life means that legacy issues can arise relatively soon after the introduction of comparatively new technologies. Moreover it would seem that that an appreciation of legacy needs to move away from a purely technological stance to admit the importance and impact of organisational issues. • Straightforward process approaches, despite their attraction to system modelers, are unlikely to take into account the various interactional subtleties involved in the actual doing of the work. In that case understanding how 'processes' may be made efficient and effective would seem to require a nuanced view of various factors, including working practice, communication and control problems, and indeed any number of complex articulations of structure, process, technology, and 'situated' knowledge.

BU	2(20)	Point of reference	• Not surprisingly, a large number of IS effectiveness measures can be found in the IS literature. What is not clear in the literature is what measures are appropriate in a particular context. In this paper we propose a two-dimensional matrix for classifying IS Effectiveness measures.
IT	2(03)	Point of reference	• (Explaining the use of a pattern language in software migration projects) The experience of four successful migration projects in five years has clearly demonstrated clearly the importance of focusing on software architecture -–(the partitioning of a system according to a specific separation of concerns -) and on achieving a strong correspondence between the key abstractions in the problem space and software components in the solution space.
IT	2(15)	Point of reference	(Largely technical: The article evaluated IMES based on its inherent quality rather than on how well it was actually used or what difference it made in the work the users were doing or the results of that work.)
IT	2(17)	Point of reference	• [This tutorial presents project management] principles and show[s] how they can be applied to the development of information systems. • Although some projects fail for technical reasons, most project failures are caused by people who ignore the principles of good project management.
IT	2(24)	Point of reference	[Reengineering patterns may be] a means of codifying and disseminating systems reengineering expertise. Through widening the definition of a legacy system to include the business process, we propose that patterns may provide a communication link between business and technology strategists that would help align their objectives and improve the sustainability of any resulting competitive advantage.

ABOUT THE AUTHOR

Steven Alter is Professor of Information Systems at the University of San Francisco. He holds a B.S. in mathematics and Ph.D. in management science from MIT. He extended his 1975 Ph.D. thesis into one of the first books on decision support systems. After teaching at the University of Southern California he served for eight years as co-founder and Vice President of Consilium, a manufacturing software firm that went public in 1989 and was acquired by Applied Materials in 1998. His many roles at Consilium included starting departments for customer service, training, documentation, technical support, and product management. Upon returning to academia, he wrote the textbook *Information Systems: A Management Perspective*. The third edition of that text was published in October 1998. His articles have appeared in *Harvard Business Review, Sloan Management Review, MIS Quarterly, Interfaces, Communications* *of the ACM, Communications of AIS, Futures, The Futurist,* and many conference transactions.

Communications of the Association for Information Systems