FINDINGS FROM A COMPREHENSIVE SURVEY IDENTIFY THE MAJOR ISSUES THAT MOTIVATE USERS TO EMPLOY TRACEABILITY PRACTICES—OR NOT.

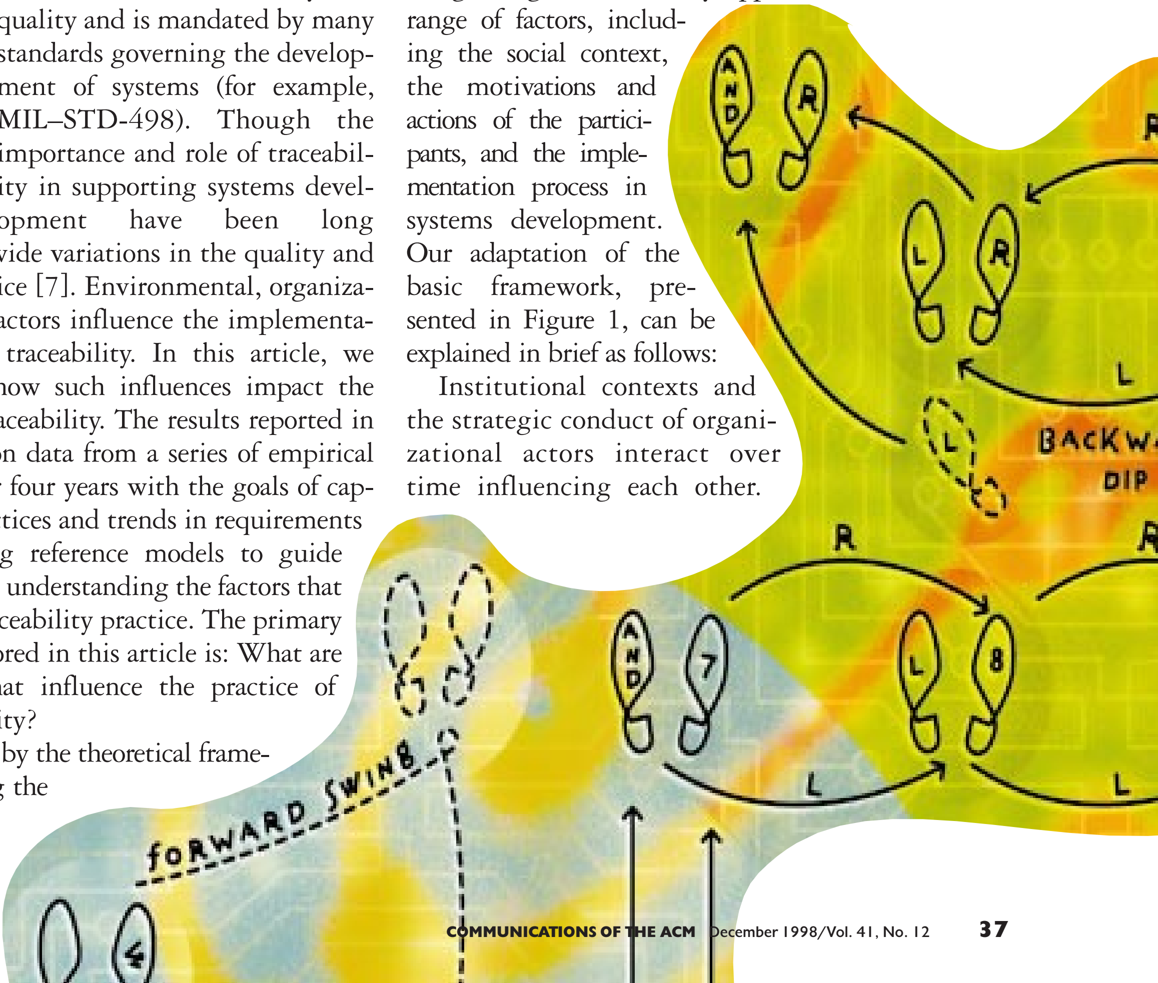
FACTORS INFLUENCING REQUIREMENTS TRACEABILITY PRACTICE

BALASUBRAMANIAM RAMESH

Requirements traceability is viewed as a measure of system quality and is mandated by many standards governing the development of systems (for example, MIL–STD-498). Though the importance and role of traceability in supporting systems development have been long recognized, there are wide variations in the quality and usefulness of the practice [7]. Environmental, organizational, and technical factors influence the implementation of requirements traceability. In this article, we identify and discuss how such influences impact the adoption and use of traceability. The results reported in this article are based on data from a series of empirical studies conducted over four years with the goals of capturing the current practices and trends in requirements traceability, developing reference models to guide improved practice, and understanding the factors that facilitate or impede traceability practice. The primary research question explored in this article is: What are the critical factors that influence the practice of requirements traceability?

The study is guided by the theoretical framework for understanding the issues around the adoption of CASE tools developed by Orlikowski [5]. This framework, developed using the grounded theory approach, considers a wide range of factors, including the social context, the motivations and actions of the participants, and the implementation process in systems development. Our adaptation of the basic framework, presented in Figure 1, can be explained in brief as follows:

Institutional contexts and the strategic conduct of organizational actors interact over time influencing each other.



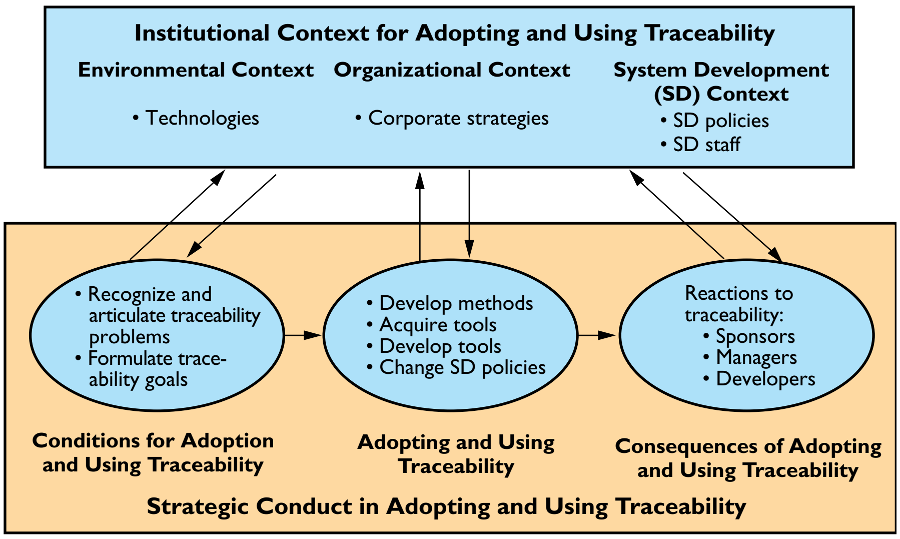

Institutional Context for Adopting and Using Traceability
Environmental Context
• Technologies
Organizational Context
• Corporate strategies
System Development (SD) Context
• SD policies
• SD staff
• Recognize and articulate traceability problems
• Formulate traceability goals
• Develop methods
• Acquire tools
• Develop tools
• Change SD policies
Reactions to traceability:
• Sponsors
• Managers
• Developers
Conditions for Adoption and Using Traceability
Adopting and Using Traceability
Consequences of Adopting and Using Traceability
Strategic Conduct in Adopting and Using Traceability

Figure 1. Factors affecting traceability practice (adapted from [5])

Characteristic	Low-end traceability user	High-end traceability user
Number of organizations in the study	9	17
Number of participants	54	84
Industries represented	U.S. government system development, program management and testing, pharmaceutical, electronics, software consulting/contracting	U.S. government system development. Program management and testing, utility, telecommunications, aerospace, electronics, automobile, software consulting/contracting
Typical complexity of system	About 1,000 requirements	About 10,000 requirements
Traceability experience level	Zero to two years	Five to ten years
User definition of traceability	Documents transformation of requirements to design	Increases the probability of producing a system that meets all customer requirements and will be easy to maintain
Main application of traceability	Requirements decomposition; requirements allocation; compliance verification; change control	Full coverage of life cycle, including user and customer; capturing traces across product and process dimensions

Table 1. Characterization of low-end and high-end usage of requirements traceability

Conditions for adopting and using traceability are influenced by environmental, organizational, and systems development (SD) contexts.

Strategic conduct involves the recognition and articulation of the traceability problem and formulation of objectives for traceability practice. These conditions lead to adoption and use of traceability, which includes the development/ acquisition of methods and tools and changes to system development policies and practices. These actions are influenced by institutional context, such as corporate strategies for traceability. The actions on adoption and use of traceability lead to the various outcomes and experiences from the key participants in the process (for example, sponsors, managers, and developers). These reactions are also influenced by the institutional context such as system development policies. (See the sidebar for details of how the study was conducted.)

Our analysis of the data quickly revealed the participants fell into two distinct groups with respect to requirements traceability practice, which we refer to as *low-end* and *high-end traceability users*. The characteristics of both groups and their representation in our data are summarized in Table 1. Low-end users view traceability simply as a mandate from project sponsors, whereas high-end users view traceability as an important component of a quality systems engineering process. The traceability practice of the two groups varies considerably. Low-end users use simple traceability schemes to model dependencies among requirements, allocation of requirements to system components, and links to compliance verification procedures. Low-end users do not capture process-related traceability information such as rationale behind various artifacts and the progressive evolution of these artifacts. High-end users employ much richer traceability schemes, thereby enabling more precise reasoning about traces. They also use traceability information in much richer ways, and emphasize the capture of process related traceability information. Detailed models representing traceability practices by these two groups of users are discussed in [11]. Though we illustrate the variation in practice by highlighting the two extremes, it should be emphasized that even an organization with a predominantly low-end practice, may have mature traceability practice in select areas and vice-versa. Our objective is not on classifying organizations, but on understanding factors that explain the range of commonly observed traceability practices.

The differences among the two groups of users are highlighted by the spectrum of views they have on the various factors influencing traceability practice. Here, we discuss how these various factors shown in Figure 1 influence the traceability practice of these two groups.

Environmental context. The technologies that support the capture and use of traceability information are equally accessible to both high-end and low-end users. High-end users, however, often tailor these tools or even develop new technologies to achieve well-articulated specific technical objectives (for example, "tight integration between various CASE tools."[1]).

Organizational context. Low-end users view traceability as a mandate from the sponsors and/or for compliance with standards. Lack of organizational commitment to a comprehensive traceability practice is a common trait among this group. High-end users, on the other hand, are strategically committed to increasing the quality of their system development process to achieve long-term improvements in organizational performance. They also recognize a comprehensive traceability practice can be used to achieve competitive advantage ("help win future contracts of similar projects or gain future cost-savings from reuse of traceability information").

System development context. Often ad-hoc practices and methodologies characterize low-end users. High-end users, however, have well-defined system development policies, such as "standardized methodologies and procedures" for traceability that are used across system development efforts. Low-end users often employ staff not involved in the development process ("outside contractors") for creating traceability documents. High-end users' commitment to traceability is evident from their use of system development staff to capture and maintain traceability information as an integral part of (or a side-benefit of) the system development process.

[1]This is a direct quote from a subject participating in a focus group or interview. Henceforth, all quotes from a subject will be included within quotation marks, but no specific reference will be made.

THE OBSERVATIONS PRESENTED HERE SHOULD BE OF INTEREST TO ORGANIZATIONS THAT SEEK TO TRANSITION FROM A LOW-END PRACTICE TO A HIGH-END PRACTICE.

Traceability Conditions

Recognize and articulate traceability problem. Low-end users view traceability documents as outputs created to satisfy sponsor requirements and/or for standards compliance. They take the view that "expensive initial investments in technology and training" needed for comprehensive traceability are inconsistent with their corporate strategies. High-end users define the scope of traceability to be much more comprehensive; such as a "trace of the process of systems development" and as a "mechanism for process improvement."

Formulate goals for traceability. Given their limited view of the traceability problem, low-end users identify simple schemes for their practice (see Table 1). High-end users consider comprehensive traceability practice as an important component of their efforts to improve and maintain process quality. Therefore, their traceability goals include not only the capture of "traceability across various products" (for example, requirements and system components), but also the maintenance of traceability information facilitating the "understanding of the processes" behind the creation of artifacts, such as design rationale.

Adopting and Using Traceability

Develop methods. Low-end users "lack a formal methodology" for traceability practice. Their goals, however, can be met by simple traceability techniques. Templates of traceability matrices required as project deliverables are created. Many high-end users realize the absence of a well-defined methodology specifying what information is needed to be captured by whom, and how it should be used, leads to unacceptable variations in the traceability information maintained by different project participants. They "define traceability schemes by developing templates and even specifying formal models into CASE tools." For example, Entity-Relationship-Attribute models supported by tools such as RDD-100 are commonly used to specify traceability models. One organization in our study realized the "wide variations in the quality, quantity, and usefulness of design rationale and effort expended" in capturing this information was a result of unclear guidelines on the form and content of this information. After the management defined templates of traceability reports, the utility of the information captured was much improved.

The traceability schemes of the two groups are markedly different in their scope and content. High-end users recognize when vertical traceability [4] between an object in one phase of the life cycle to the next is lost, its traceability to objects in the subsequent phases is also lost. Low-end users commonly face this problem. For example, they capture very little prerequirement-specification traceability, that is, traceability that addresses the question: Where do requirements come from? [2]. ("We have not had sensitivity to traceability at requirements conception. Traceability has been a fallout of the testing."). High-end users try to avoid such breakdown by maintaining traceability across all phases. Their prerequirements-specification traceability efforts are markedly more elaborate. For example, some high-end users conduct their own "independent analysis of the organizational needs to extract rationale behind requirements." They compare these with the requirements document to identify discrepancies and seek clarifications.

Low-end users do not maintain detailed accounts of the processes by which various artifacts are developed. High-end users, on the other hand, recognize that critical elements of the development process should be clearly traced to their stakeholders. ("I'm sure that I'm going to want to look back in the future and ask myself who made certain decisions or where decisions came from.") For example, an organization requires that each requirement have an owner who is responsible for its justification and evolution.

Low-end users often create static documents (traceability matrices) that do not get updated as the system evolves and, therefore, lose much of their usefulness soon after their creation. High-end users recognize the need for maintaining dynamic traceability information that reflects the current status of the system and, when feasible, generate traceability documentation that can be "derived at any point in the life cycle as a by-product of the system development effort." However, when products are delivered across inter- or intraorganizational boundaries that use different environments, (say, at the end of project life cycle phases), maintenance of such living traceability documents become very difficult, if not impossible.

Acquire tools. Commercial tools that support traceability differ widely in their functionality and scope. Many simple traceability tools, widely used by low-end users, are based on relational databases. Relevant information (such as requirements, test plans) is entered or imported into these to produce traceability matrices. As they are often decoupled from the development environment, they have "very limited utility in capturing dynamic traceability information."

Develop tools. High-end users prefer to use tools that are "embedded within the development environment." However, many popular requirements traceability tools address only limited aspects of the system development life cycle (DOORS for Requirements Management). Further, complex projects require the "use of a variety of tools" in systems development. To overcome the problem of incompatibility among CASE tools, and to supplement their traceability capabilities, many have developed in-house tools and utilities.

Change system development policies. Factors in this group include the attitude toward traceability costs, selectivity in trace capture and maintenance, and organizational implementation strategies.

Low-end and high-end users differ strongly in their attitude toward traceability costs. Traceability is considered by low-end users as an "expensive overhead." As a nonfunctional requirement, traceability tends to slip to the end of the project life cycle and is one of the first things to be eliminated or scaled down when a funding crunch comes along. ("If nobody pays you to document and trace, then you don't do it.")

High-end users take a life cycle view of costs and benefits to justify their comprehensive traceability practice. Cutting traceability in response to resource constraints is considered inappropriate. ("When you lose traceability, you also lose some management decision aids, such as the ability to perform impact analysis.") They view traceability as an auditing

Category	Concept		Facilitating Characteristics	Impeding Characteristics
Environmental context	Technologies		Tailor or develop new technologies	Inability to effectively use available techologies
Organizational context	Corporate strategy		Organizational commitment for quality system development	Traceability as a mandate
System development context	System development policies		Standardized methodologies	Ad-hoc practices
	System development staff		Project staff	External staff
Conditions for adoption and use of traceability	Traceability problem		Mechanism for process improvement	Required overhead
	Traceability goals		Capture process/product dependencies	Sponsor/standards compliance
Adopting and using traceability	Develop methods		Well defined, across all phases, links to stakeholders	Ad-hoc, select activities
	Develop/acquire tools		Tailored, embedded in system development environment	Stand-alone Incompatibility among tools
	Change system development policies	Costs	Life cycle view of costs	Treated as an overhead
		Scope	Selective capture	Uniform capture
		Implemen-tation strategy	Incremental adoption Adequate training and support	No clear strategy Inadequate training and support
		Human resource policies	Tangible and intangible benefits Use of traceability for quality assurance	Inadequate incentives and protection Use of traceability for performance appraisal

Table 2. Factors facilitating and impeding traceability practice

framework necessary "for monitoring resource allocation and use."

Another difference concerns the need for selective trace capture. Low-end users capture traceability information uniformly for all requirements. While this is feasible with simple traceability schemes, it may prove very expensive with elaborate traceability capture. High-end users often recognize that "all requirements are not equal" in terms of their significance or criticality. It may be unnecessary or even undesirable—considering the overhead involved in maintaining traceability—to link every requirement with every output created during the systems design process. Many consider it essential to maintain detailed traceability only from mission-critical requirements. ("Traceability is feasible only for critical aspects of the project to keep the costs under control and get comparable benefits.")

Low-end users face another critical inhibitor: resistance from many developers who think the additional workload may adversely impact their individual productivity. High-end users adopt a few successful implementation strategies to overcome this problem. First, the technology may be inserted into a pilot project with the explicit goal of process improvement. The successful implementation of traceability (rather than the traditional measures of productivity) is identified as an important measure of project success. Such a loss leader strategy helps achieve buy-in from the participants. Second, a core group of stakeholders is trained in traceability methodology and tools to act as internal consultants to help in technology transition in a phased manner. Besides shielding the average user from unstable and evolving technology, they also participate in peer reviews to provide the much-needed guidance on the quality of traceability efforts.

The importance of proper human resource policies was uniformly recognized by all the focus groups. The following issues were identified as very important by all but two focus groups: Often the person who captures a piece of traceability information (like the requirements engineer) is not the one who may use it later (for example, the maintainer). If participants perceive traceability as a mechanism to explicitly document their expertise, they may fear for their job security.

Efforts toward maintaining detailed traceability are not adequately rewarded by low-end users. Therefore, they tend to view traceability as an unnecessary overhead. High-end users recognize the importance of the issues we've mentioned in providing both tangible and intangible incentives ("providing adequate allowances in task schedules and costs for the overhead"). They also recognize the need to educate the participants on the criticality and usefulness of traceability across the life cycle. Project meetings and reviews are used to bring together the potential producers and consumers of traceability information to highlight the "criticality of efforts expended in traceability."

Many high-end users are keenly aware of the need to assure the participants that knowledge-sharing among project team members provides many benefits. Formal mechanisms such as peer reviews are used to "provide valuable feedback and encouragement" to individuals. Further, by providing access the organizational memory of traceability information (such as critical issues and design decisions) and rewarding contributions, individuals are encouraged to participate.

HOW THE STUDY WAS CONDUCTED

High-end users also indicated the importance of proper use of traceability information. The use of accountability information as a means for performance appraisal is considered inappropriate. ("Traceability should not be used to threaten people with.") A technique employed by a high-end organization to minimize the risk of such misuse is to have accountability for the critical elements (such as design decision) assigned to an entire group after providing the group with the opportunity to review the outputs of its members periodically.

Consequences of Adopting and Using Traceability

The most critical factors facilitating and impeding traceability practice are summarized in Table 2. The reactions of the key organizational participants are influenced by these factors.

Sponsor reactions. Low-end users suggest that some project sponsors may not fully appreciate the benefits of traceability beyond the need to comply with standards. The static traceability documents produced by most low-end user become obsolete even as they are delivered. High-end users indicate that project sponsors often "have to be educated about the consequences" of lack of adequate funds to support comprehensive traceability, especially in the early phases.

Manager reactions. Managers of low-end users claim they do not derive much benefit from their simple traceability practices. It is viewed as a necessary evil and is attempted only when the deliverables are due, thereby severely limiting its usefulness. High-end user managers, in contrast, are committed to traceability as a mechanism for improving and maintaining the quality of the systems development process and see strategic benefits of incorporating traceability, even when it is not required by the project sponsors. These include the ability to develop realistic cost proposals and gaining competitive advantage in building similar systems due to "savings from using lessons learned database of critical issues and rationale." Managers of one organization that moved from Level 1 to Level 3 of the SEI CMM strongly believe their comprehensive traceability practice ("well beyond the narrow interpretation of CMM requirements") was an important factor in achieving this goal [10].

Developer reactions. The lack of interest in traceability among low-end developers is easily explained by the "lack of organizational commitment, management support," adequate tools, methodologies, and training. Even when some individuals capture traceability for personal use (design notebooks as "memory joggers"), the organization does not derive much benefit from such efforts. High-end developers have a variety of motivating factors to view traceability positively. Adequate incentives are provided in an atmosphere of knowledge sharing and growth. Therefore, traceability is not viewed as a threat to job security. The support structure provided for learning the methodology, tools, and techniques, may enhance their marketability and value to the organization. However, individual attributes, such as their career goals and system development experience will also be important factors determining the disposition of individual developers.

Implications for Research and Practice

We have focused on the role of institutional context and the strategic conduct in explaining the differences in traceability practice across system development efforts. By contrasting two extremes of practice—low-end and high-end—we have attempted to provide an understanding of these factors affecting traceability practice. Further studies are required to examine and elaborate on individual factors to statistically validate our observations. Our work complements the current literature on traceability in a number of ways. Most of the work on traceability is aimed at developing tools and methodologies for supporting select aspects of the traceability problem [6]. For example, Gotel and Finkelstein [2] use contribution structures, a rich system of stakeholder roles for identifying the origins of requirements. Similarly, Yu and Mylopolous [12] use dependencies between stakeholders as an important factor in the development of requirements. We present the larger environmental and organizational context which determines the success of traceability.

The observations presented here should be of interest to organizations that seek to transition from a low-end practice to a high-end practice. Significant benefits of such a transition may include increased process maturity and lower life cycle costs [10].

Our results should also have implications for requirements traceability tool developers. First, the development of tools to provide traceability across different development environments is considered very important by high-end users. Large projects often need to dynamically reference and link information created and maintained in a variety of tools and platforms [7]. Second, in the absence of automated tools, traceability will not only be error-prone and time-consuming, but may be impossible to maintain (due to the "extensive volume of data"). Whereas some tools provide limited support for capturing traceability as a by-product of development as desired by high-end users, their usefulness is limited by their predefined process models. Further, most current tools used in practice do not capture precise semantics of the traceability information, limiting their ability for automated reasoning—a capability desired by high-end users. Though there are very few process-centered environments commercially available, recent research [9] illustrates the feasibility to define and enact process steps that will automate the creation and maintenance of traceability information. TOORS [8] and PRO-ART [9] are examples of recent research efforts that support automated reasoning to enhance the usefulness of traceability information. C

References

1. Eisenhardt, K.M. Building theories from case study research. *Acad. Manage. Rev. 14,* 4 (Oct. 1989), 532–550.
2. Gotel, O and Finkelstein, A. Extended requirements traceability—Results of an industrial case study. In *Proceedings of the 3rd International Symposium on Requirements Engineering.* (Annapolis, Md., Jan. 1997), 169–178.
3. Morgan, D.L. *Focus Groups as Qualitative Research.* Sage Publications, Newbury Park, Calif., 1988.
4. Nejmeh, B.A., Dickey, T.E. and Wartik, S.P. Traceability technology at the software productivity consortium. G.X. In Riter, (Ed). *Information Processing* '89. Elsevier Science Publishers, New York, (1989).
5. Orlikowski, W. CASE Tools as organizational change: Investigating incremental and radical changes in systems development. *MIS Q.* (Sept. 1993), 309–340.
6. Lindvall, M. and Sandahl, K. Practical Implications of Traceability. *Software—Practice and Experience 26*, 10 (1996), 1161–1180.
7. Palmer, J.D. Traceability. *Software Requirements Engineering.* R.H. Thayer. and M. Dorfman (Eds). IEEE Computer Society Press, New York, 1997.
8. Pinheiro, F. and Goguen, J. An object-oriented tool for tracing requirements. *IEEE Softw.*, (Mar. 1996), 52–64.
9. Pohl, K., Dömges, R. and Jarke, M. Towards selective, model-driven trace capture. In *Proceedings of the 7th International Conference on Advanced Information Systems Engineering.*(Barcelona, 1997).
10. Ramesh, B., Stubbs, C., Powers, T., and Edwards, M. Requirements traceability—Theory and practice. *Annals of Softw. Eng. 3,* (1997), 397–415.
11. Ramesh, B. and Jarke, M. *Towards Reference Models for Requirements Traceability*. Technical Report, Georgia State University, 1998.
12. Yu, E and Mylopoulos, J. Understanding 'why' in software process modeling. In *Proceedings of the 16th International Conference on Software Engineering.* (Sorrento, Italy, 1994), 159–168.

Balasubramaniam Ramesh (bramesh@gsu.edu) is an associate professor of Computer Information Systems in the College of Business Administration at Georgia State University.

This work was supported in part by grants from the Office of Naval Research/NSWCDD and the National Science Foundation.