



Journal of Management Information Systems

Publication details, including instructions for authors and subscription information:
<http://www.tandfonline.com/loi/mmis20>

An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques

Glenn J. Browne, Michael B. Rogich
Published online: 09 Jan 2015.

To cite this article: Glenn J. Browne, Michael B. Rogich (2001) An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques, *Journal of Management Information Systems*, 17:4, 223-249

To link to this article: <http://dx.doi.org/10.1080/07421222.2001.11045665>

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution,

reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at <http://www.tandfonline.com/page/terms-and-conditions>

An Empirical Investigation of User Requirements Elicitation: Comparing the Effectiveness of Prompting Techniques

GLENN J. BROWNE AND MICHAEL B. ROGICH

GLENN J. BROWNE is Assistant Professor in the area of Information Systems & Quantitative Sciences at Texas Tech University. He received his Ph.D. in Information Systems and Decision Sciences from the Carlson School of Management at the University of Minnesota. His research interests include information requirements determination, conceptual modeling, and behavioral decision-making processes. His articles have appeared in *Management Science*, *Organizational Behavior and Human Decision Processes*, *IEEE Transactions on Systems, Man, and Cybernetics*, the *Journal of Systems and Software*, and other journals.

MICHAEL B. ROGICH is Director of Online Learning at Saint Leo University. He received his Ph.D. in Information Systems from the University of Maryland, Baltimore. His research focuses on improving the information systems development process. Before entering academia, he spent ten years developing and managing the development of large-scale information systems in the financial services industry.

ABSTRACT: Eliciting requirements from users and other stakeholders is of central importance to information systems development. Despite this importance, surprisingly little research has measured the effectiveness of various requirements elicitation techniques. The present research first discusses theory relevant to information requirements determination in general and elicitation in particular. We then develop a model of the requirements elicitation process. This model and its underlying theory were then used to construct a new requirements elicitation prompting technique. To provide a context for testing the relative effectiveness of the new technique, two other questioning methodologies were also operationalized as prompting techniques: (1) the interrogatories technique, which involves asking “who,” “what,” “when,” “where,” “how,” and “why” questions; and (2) a semantic questioning scheme, which involves asking questions based on a theoretical model of knowledge structures. To measure the usefulness of the prompting techniques in eliciting requirements, a set of generic requirements categories was adapted from previous research to capture requirements evoked by users. The effectiveness of the three methods in eliciting requirements for a software application was then tested in an experiment with users. Results showed that the new prompting technique elicited a greater quantity of requirements from users than did the other two techniques. Implications of the findings for research and systems analysis practice are discussed.

KEY WORDS AND PHRASES: information systems development, prompting techniques, requirements elicitation, systems analysis

THE ELICITATION OF REQUIREMENTS FROM USERS is a critical activity in systems development. Ideally, requirements would flow plainly and clearly from users to systems analysts. However, for a variety of cognitive, communicative, and motivational reasons, the information ultimately received and understood by analysts is generally incomplete [3, 15, 50, 57, 58]. As a result, actions taken that rely on such information are often unsuccessful. Thus, improvements in the elicitation of requirements hold much promise for improving systems development efforts.

Though information must be sought from users and other interested parties throughout the systems development lifecycle, of particular importance is information elicited during what is usually referred to as information requirements determination. Because the requirements determination stage occurs early in the systems development process, improvements have a ripple effect throughout the lifecycle. A complete and accurate collection of user requirements sets the stage for an efficient development process that increases the likelihood of a successful organizational system [15].

To improve the requirements determination process, this paper first discusses theory relevant to the process. We then analyze the requirements elicitation task and develop a detailed model of this task. Difficulties in evoking information are then discussed and used to motivate the development of a new theory-driven prompting technique for analysts to use in eliciting requirements from users. Next, we operationalize two existing questioning methodologies into sets of elicitation prompts. To be able to measure the relative effectiveness of requirements elicitation efforts, we then extend an existing categorization scheme for capturing and organizing information evoked by users. Finally, we test the new prompting technique and the two other elicitation methodologies. This is the first empirical test of the effectiveness of prompting methods in eliciting information from users in the information systems domain.¹ We conclude with a discussion of the findings and implications for researchers and practitioners.

Background

Requirements Determination

THE DIFFICULTIES ENCOUNTERED IN DEVELOPING INFORMATION SYSTEMS are well documented. Despite good faith efforts by organizations, analysts, and users, a majority of systems are either abandoned before completion or fail to meet user requirements, costing organizations more than \$100 billion a year in the United States alone [18, 51]. A principal reason that systems do not meet user expectations is the failure of the development process to yield a complete and accurate set of requirements [14, 15, 59]. Requirements determination is the process of gathering and modeling information about required functionality of a proposed system by a systems analyst, and has been widely recognized as the most difficult activity of information systems development [5, 12, 15, 24, 34, 53, 62, 66]. Improvements in requirements determina-

tion methods can have a dramatic impact on both the effectiveness of systems and the efficiency of the development process [57, 59].

Broadly speaking, requirements determination can be viewed as a three-step process: (1) information gathering, during which requirements are elicited from users by analysts; (2) representation, during which the evoked requirements are modeled in a physical form by the systems analyst; and (3) verification, during which the analyst verifies with the user that the requirements modeled are indeed correct [30, 60]. The present research is concerned with the elicitation portion of information gathering, particularly ways in which such elicitation can be improved.

Although a number of strategies exist for eliciting system requirements [15], the most popular strategy is for an analyst to interview the users of the proposed system [1, 7]. Structured interviews are often recommended (e.g., [27, 64]), but surprisingly little guidance is offered regarding the content of such interviews [37]. Typically, the lack of methods has forced the analyst to rely solely on his or her ingenuity in talking with users. Further, the lack of tools for conducting structured interviews often results in analysts asking the wrong questions or omitting important questions altogether [63].

The principal method used in structured interviews is to ask the people who will ultimately use the new system about the procedures they follow in performing their tasks and the types of information they need to perform their jobs [15]. The asking method is typically operationalized as a set of directed questions. A directed question, most generally, is a question designed to cause a user or domain expert to attend to and focus on a particular type of information [20]. The use of directed questions is intuitively appealing because of the natural inclination to ask people what they want or need in a situation. Thus, it is not surprising that directed questions have been used in domains ranging from general problem solving [28] to risk analysis [19] to decision analysis [61]. Recently, approaches to directed questions have been developed that attempt to overcome cognitive limitations in evoking information [6]. This prior research is used as the basis for directed questions developed in the current study.

Before developing questions to aid analysts in eliciting requirements from users, it is useful to analyze and model the requirements elicitation task. A theoretically sound task model can help determine the types of prompts that should be developed to improve the requirements determination process. Therefore, we next analyze the task of eliciting requirements from users.

A Model of the Requirements Elicitation Task

Although many researchers have studied requirements determination, actual models of the task of eliciting requirements are few. There are many advantages of such a model. A model of the elicitation task will allow the analyst to understand what requirements are and how they might be captured, in addition to giving him a clearer understanding of what his goals should be. Further, the proposed model will help the analyst see requirements not merely as the documentation of data elements in a business process, but rather as the identification of goals, assumptions, opinions, and

desires of users [7, 41, 65]. Using theories from human problem solving, cognitive psychology, and information systems development, a model of the requirements elicitation task was constructed. This model appears in Figure 1. We now describe the model in detail.

The stimulus for systems development is typically an organizational problem or opportunity. Problems may be usefully analyzed in terms of organizational goals, business processes, tasks that must be performed within the processes to achieve the goals, and information (data) necessary to inform task behaviors [65]. We represent these elements in a pyramid structure in the model for two reasons. First, it is useful to think of the elements hierarchically, with general-level goals at the top and specific information at the bottom. Second, we believe that in a systems development effort of any size, there will be fewer goals than processes, fewer processes than tasks, and fewer tasks than information. Thus the pyramid structure is a useful means for organizing and describing types of requirements.

The user and the analyst each conceptualize the problem task environment. These conceptualizations are referred to as problem spaces [38]. When engaged in evoking requirements, the user's problem space contains information recalled from long-term memory in response to needs in working memory. Requirements evocation is mediated by such factors as failure to recall information from long-term memory, cognitive biases, and inability to articulate routinized procedures that have become "automated" [43, 44].

To gather requirements successfully, the analyst must elicit from the user both a conceptualization of the problem and a visualization of a preferred future organizational environment. Although it is often difficult for the user to recall details of the current environment, it is even more difficult to envision the future preferred organizational environment [67]. Based on what he elicits from the user, the analyst forms an initial conceptualization of current and future states of the organization by organizing the user's responses. To improve his understanding and to collect missing information, the analyst uses various types of prompts to stimulate the user's thought processes further (illustrated in the model by arrows between the two problem spaces). As the analyst begins to build an understanding of the user's needs, he commits psychologically to partial solutions to the problem [49]. These partial solutions undergo extensive mental simulations and reformulations that are eventually terminated by the analyst's use of evaluative stopping rules [22, 40]. When the analyst has what he believes is enough information, he records the information on a requirements document. The requirements document, which includes information elicited from users and gathered from other sources, represents a description of a future state of the world that, when implemented, includes a system that is aimed at enabling the organization to achieve the goals identified.

This model of the requirements elicitation task is designed to guide research and practice toward an understanding of requirements elicitation, which is a central part of the overall requirements determination process. Many aspects of the model require testing and validation. In the current study, we focus on developing prompts to aid the analyst in eliciting requirements from the user about needs for the proposed

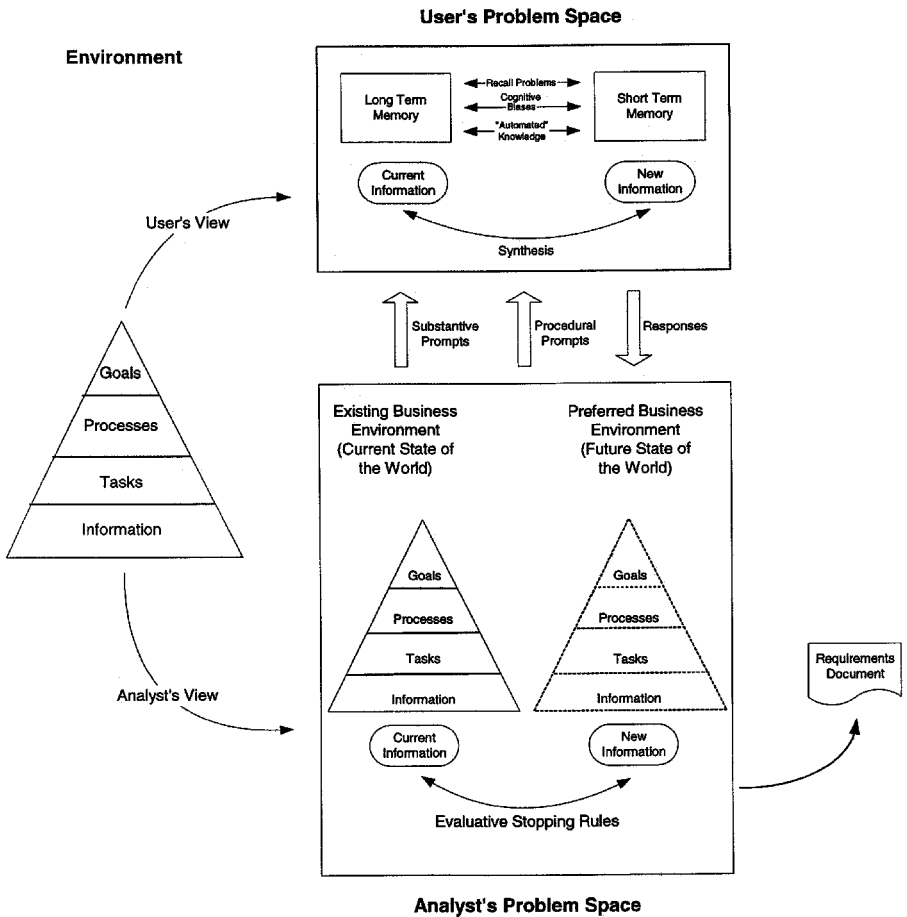


Figure 1. Requirements Elicitation Task Model

information system. To motivate the types of prompts that may be helpful in eliciting users' knowledge, we now describe difficulties that arise in the requirements elicitation process.

Difficulties in Requirements Elicitation

Difficulties in requirements elicitation occur for a number of reasons. Among the difficulties listed by Davis [15], three are relevant in the current research: (1) cognitive issues resulting from constraints on humans as information processors; (2) problem structuring issues resulting from the variety and complexity of information requirements; (3) communication issues resulting from the complex patterns of interaction among users and analysts in defining requirements. These problems are all illustrated explicitly or implicitly in Figure 1. Cognitive problems are represented by the factors that mediate the user's thinking—for example, recall problems. Problem

structuring issues are most apparent for analysts, and are represented by the requirements pyramids that must be conceptualized for both the existing and preferred business environments. Communication issues are apparent from the dialogue arrows between the user's problem space and the analyst's problem space.

Although all of these difficulties are important in the requirements elicitation process, in the present research we focus on cognitive problems of users. Of particular concern in the elicitation of requirements are the obstacles shown in Table 1. These obstacles are drawn from empirical work in cognitive psychology and information systems development. Excellent discussions of many of these obstacles can be found in Stacy and Macmillan [50] and Valusek and Fryback [58]. For present purposes, we focus on strategies to overcome the obstacles.

The theory underlying development of the strategies to overcome cognitive obstacles was drawn primarily from the fields of rhetoric and behavioral decision-making. One theory of specific interest is the theory of practical reasoning, which is the means people use to express information and reason about problems in everyday life [56]. Research in practical reasoning has considered several mechanisms for expressing information, including argument and strategy types [11, 39]. In the current study, we are interested in strategy types that can aid users in evoking information. Strategies are means for manipulating, limiting, or combining assertions in useful ways. Types of strategies identified by researchers include building scenarios, conditionalizing, elaborating with instances, hedging, and generating counterarguments [11, 29]. Table 2 provides examples and descriptions of a number of different strategies. Strategies have been shown to be useful as prompts because they elicit information that people otherwise would not evoke in attempting to solve problems [6]. These strategies are used in the prompting technique developed below.

Prompting Techniques for Improving Requirements Elicitation

Previous research has distinguished between two general types of prompting techniques: context-dependent techniques and context-independent techniques [6]. Context-dependent techniques are directed questioning schemes designed for specific contexts. Such techniques have generally attempted to use context to help users focus on specific goals for the system and to keep their reasoning confined to narrow issues [36, 54]. The advantage of such techniques is that they are usually more powerful than context-independent techniques because they take advantage of the context in which a system is being developed [38]. However, there are at least two weaknesses of such techniques. First, different sets of directed questions need to be created for each type of systems development effort, since the schemes are not widely generalizable. Second, significant substantive expertise in the users' domain must be utilized by the analyst to construct such questions. In the absence of such expertise, the context-dependent questioning scheme is likely to be ineffective [2]. Further, even significant domain knowledge does not guarantee that requirements elicitation will be effective. An analyst with domain knowledge may fail to gather sufficient requirements either

Table 1. Cognitive Obstacles to Requirements Elicitation

Nature of Obstacle	Impact of Obstacle	Strategies to Overcome
Short-term memory	Users can only focus on a limited amount of information {43}.	Generating arguments, feedback, summarization.
Constructive nature of long-term memory	Recall problems {43}.	Scenario building, generating arguments, elaborating with instances.
Bounded rationality/satisfying	Premature use of stopping rules, use of faulty heuristics {46}.	Scenario building, feedback, counterargument.
Automaticity	Users unable to recall procedural details of how they perform tasks {43}.	Scenario building, counterargument.
Faulty reasoning	Reasoning fallacies, inappropriate arguments, not enough arguments {39}.	Generating arguments, feedback, counterargument.
Cognitive biases	Users' responses biased because of use of cognitive heuristics {15}.	Scenario building, counterargument.

Table 2. Strategy Types for Eliciting Requirements (Adapted from [11, 26, 29])

Strategy	Description	Example
Scenario Building	Asking a user to imagine or construct a scenario in his domain, and respond as he would in that situation.	Describe the most unusual customer you ever had. How did you respond in that situation?
Conditionalizing	Using an "if-then" clause to modify an assertion to limit or clarify its applicability.	If the project is completed as planned, then what does that mean for the customer?
Elaborating With Instances	Asking a user to illustrate a point by providing examples.	Can you provide some examples of what you mean?
Hedging	Asking a user to design contingency plans or fallback positions.	What would you do if your solution did not work?
Generating Counterarguments	Asking a user to argue against the conclusion he first reached.	Why might the system not work as well as you say it will?
Generating Arguments	Asking a user to make more arguments favoring his position, or different kinds of arguments, e.g., causal arguments, similarity arguments, or analogy arguments.	Can you think of any analogies that would help clarify what you are saying?
Feedback	Asking a user for feedback or providing him with feedback on what he has said, either verbally or in the form of notes or diagrams.	I've written down everything you have said, and here is a copy.
Summarization	Asking a user to summarize what he has said, or providing a summarization for him.	Can you summarize everything you have said thus far?

because he does not understand theories that lead to good question construction or because he has no systematic elicitation methodology.

The second type of prompting technique is one independent of context. Such techniques have significant potential utility since they are exportable from task to task. The weakness of the techniques is that their generality typically leads to less power than good context-dependent questioning schemes [38]. However, as noted, constructing context-dependent questions requires significant expertise in the application domain. Since systems analysts will often (if not usually) be assigned to analyze business processes for which their substantive knowledge is limited, context-independent questions can provide an excellent basis for interviewing users. This is not to suggest that such questions can be used alone to gather requirements, any more than interviewing alone can be used. Further, the questions will likely need to be adjusted and supplemented as appropriate given the nature of the process being analyzed. However, analysts typically have little basis for knowing what to ask, and context-independent directed questions can provide that basis.

We used the model of the requirements determination task developed above and its underlying theory to develop a context-independent prompting technique to aid analysts in eliciting information from users. We term the technique the “task characteristics technique” because it is based on an analysis of the requirements elicitation task. The technique is operationalized as a set of directed questions. The questions appear in Table 3, and include two types of prompts: substantive prompts and procedural prompts. Substantive prompts are aimed at eliciting specific types of requirements. Procedural prompts utilize strategies drawn from the theory discussed above, and are designed to elicit information that subjects might otherwise not evoke due to cognitive obstacles (such as forgetting, cognitive biases, etc.) [15, 50].

Both types of prompts are intended to help users and analysts overcome cognitive problems associated with the requirements elicitation task. The substantive prompts are based on a theoretical account of the requirements elicitation task, and so should stimulate users’ memory structures in ways that improve recall of knowledge relevant to system requirements. The procedural prompts are based on theory and empirically demonstrated reasoning strategies people use, and therefore should result in a more complete set of requirements than would otherwise be evoked. These procedural prompts are designed to help users overcome a variety of cognitive biases. For example, the prompt “What kinds of things can people do now that they might not be able to do when using the system?” is intended to cause users to reexamine their assumptions about the system and to play the devil’s advocate. Such counterargument prompts have been shown to reduce judgmental biases [29]. As another example, a procedural prompt might ask a user to summarize his thoughts, which can help overcome the limitations of short-term memory. Finally, a procedural prompt can ask a user to create a scenario—that is, to tell a story—about how he performs a task (a similar strategy is a “use case,” cited by some authors (e.g., [64])). Such scenarios have been shown to help people articulate information that they would otherwise not evoke [6]. All such prompts can cause users to evoke information that may improve the completeness of the system design and anticipate problems with the system as

Table 3. Task Characteristics Prompting Technique

What would your customers want the system to do? Substantive Prompt	What people or departments must be involved to support the employees' use of the system? Substantive Prompt
Why would your customers not want to use the system? Procedural Prompt—Causal Counterargument	Describe in detail the tasks that these people or departments must do. Substantive Prompt
What can be done to overcome these negatives? Procedural Prompt—Causal Counterargument	What feedback must the system provide to assist in performing these tasks? Substantive Prompt
What would your employees want the system to do? Substantive Prompt	Can you think of a situation in which the customer would have to make a decision or choice when using the system? Procedural Prompt—Scenario Building
Summarize everything you want the system to do. Procedural Prompt—Summarization, Feedback	What kinds of things can people do now that they might not be able to do when using the system? Procedural Prompt—Casual Counterargument
What must the customer do to use the system? Substantive Prompt	What information must a customer supply to the system to be able to use it? Substantive Prompt
What must the employees do to use the system? Substantive Prompt	What information must the system supply to the customer? Substantive Prompt
Can you think of a situation in which the customer would have a problem using the system? Procedural Prompt—Scenario Building	What information must the employees supply to the system to be able to use it? Substantive Prompt
What can be done to overcome these problems? Procedural Prompt—Causal Counterargument	What information must the system supply to the employees? Substantive Prompt
Summarize the steps for using the system. Procedural Prompt—Summarization, Feedback	
What people or departments must be involved to support the customer's use of the system? Substantive Prompt	

currently envisioned. Such information is often not considered, resulting in systems that are underspecified or mistakenly conceived.

To provide a context for testing the relative effectiveness of the task characteristics technique, we also operationalized two other context-independent questioning methodologies. One such methodology is the “interrogatories” technique, which relies on questions beginning with “who,” “what,” “where,” “when,” “why,” and “how” [4, 10, 16]. The interrogatories method has been recommended for use in requirements determination and other contexts in which the generation of information is important [10]. In the case of requirements determination, an analyst can use these interrogatories as syntactic building blocks for constructing questions. However, each specific question relies on the ingenuity and ability of the analyst, and the only guidance for question construction is the interrogatories themselves. Because the interrogatories technique is often used in practice, but offers no significant guidance to analysts, it is used as a control group in the present research (and termed the “syntactic prompting technique”). Questions created for the current research using these interrogatories are shown in Table 4.

Another context-independent questioning method that has been proposed is a “semantic” method developed by Lauer [31, 32]. Relying in part on speech act theory [13] and question answering classifications [23, 33], Lauer created a questioning method that attempts to take advantage of people’s organization of knowledge. His questions are classified according to events, states or conditions, actions, agents, and goals. The usefulness of Lauer’s questions in helping analysts construct data-flow diagrams (DFDs) has been investigated twice—once by Lauer himself [32] and more recently by Marakas and Elam [35]. Although these studies found that Lauer’s questions are useful for helping analysts create DFDs, the studies left actual requirements elicitation as a black box because they did not measure the information elicited. The present research uses Lauer’s questioning method as the basis for a treatment group. Questions created for the present study using Lauer’s method appear in Table 5.²

In summary, methods used for eliciting systems requirements from users have in general been ad hoc, and recommendations in the literature have been few. The task characteristics technique developed in the present research is an attempt to provide a technique that is relatively generalizable, but that also trades some generality for some power. It is generalizable because it can be used for various types of systems development efforts. It has power because the questions are based on a theoretical model of the requirements elicitation task. Though not dependent on the context of the system being developed, it is dependent on the context of requirements elicitation generally and thus takes advantage of similarities across systems development efforts. It is hypothesized that this trade-off will result in improved requirements elicitation.

Generic Requirements Categories

Once requirements have been elicited from users, analysts still face several immediate problems. One difficulty is the way in which elicited requirements should be captured. To overcome this problem, a coding scheme is necessary to organize information

Table 4. Syntactic Prompting Technique (Adapted from [10])

Who uses the system?	Who is affected by the system?	Who affects the system?
What kinds of things do they do?	What kinds of things affect them?	What kinds of things affect the system?
Where do they do them?	Where are they affected?	Where do they affect the system?
When do they do them?	When are they affected?	When do they affect the system?
Why do they do them?	Why are they affected?	Why do they affect the system?
How do they do them?	How are they affected?	How do they affect the system?

Table 5. Semantic Prompting Technique (Adapted from [32])

Goals		Events
What are the system goals?		What events affect the system?
How is each goal attained?		What are the consequences of each event occurring?
Why is each goal important?		What causes each event to occur?
What indicates that each goal is achieved?		What goal does each event fulfill?
Agents		States or Conditions
Can you name a person or department involved with the system?		What states or conditions affect the system?
What role does each play?		What causes or enables each state?
What are his or her goals?		What are the consequences of each state being present?
What agent has opposing goals?		What goal does each state support?
Actions		
Can you name the actions involved in the system?		
How does a person perform each action?		
What prevents a person from being able to perform each action?		
What goal(s) does each action satisfy?		

before system diagramming is begun. A second problem, noted above, is when the analyst should stop eliciting requirements. That is, when does the analyst have all the requirements he will need to analyze the business process and design the system? The choice of stopping rule is a complex question that will not be addressed directly here. Instead, we believe that a list of requirements categories can serve as a useful surrogate for more sophisticated stopping rules. When an analyst has elicited users' knowledge that has been coded into most or all of the requirements categories, he may feel confident that he has at least a starting basis for analysis and design.

To develop a set of generic categories that can be used by analysts for requirements determination, we have expanded upon the "Taxonomy of Generic Requirements" scheme developed by Byrd et al. [7]. Generic requirements categories are context-independent categories that can be used to evaluate the results of any requirements determination effort. A context-independent categorization scheme is of great potential value, since it eliminates the need to create different categories to evaluate the requirements for each new information system application. The proposed categories appear in Table 6, and include general level categories and subcategories. The general level requirements categories reflect the general types of knowledge that analysts need to understand from the business environment, as shown in the requirements determination task model depicted in Figure 1: goals, processes, tasks, and information [65].

The first general level category of requirements, goal level requirements, consists of organizational or strategic issues relevant to the system design. These requirements are concerned with an understanding of the overall context in which the system is being developed. Goal level requirements include a description of the current state of the organization, the future preferred state, and the means and strategies that might be used to achieve the future state [47, 48].

In addition to the goal level requirements, processes that must exist to support the organizational goals must be determined. A business process can be defined as "a series of steps designed to produce a product or service" ([45], p. 45), or "a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer" ([25], p. 35). Various types of processes are captured in the subcategories.

The third level of requirements is task level requirements. Business processes are accomplished by individuals or groups of people performing tasks. Job task requirements are behavioral in nature and consist of the following: job design objectives, work organization objectives, individual role and responsibility assumptions, and organizational policies [15]. Subcategories at the task level are designed to reflect these types of requirements.

To perform their organizational tasks, people need information. Consequently, the fourth general level category is information requirements. These requirements concern inputs, outputs, stored data, and the manipulation of data.

For systems analysts, these categories provide a template for evaluating the requirements determination effort. This is important, since research indicates that most analysts' efforts are ad hoc, inefficient, and highly variable from system to system [57,

Table 6. Generic Requirement Categories (Adapted from [7])

Generic Requirement	Description
Goal Level Requirements	
Goal State Specification:	Identifying the particular goal state to be achieved.
Gap Specification:	Comparing existing and desired states.
Difficulties and Constraints:	Identifying factors inhibiting goal achievement.
Ultimate Values and Preferences:	Stating the final ends served by a solution.
Means and Strategies:	Specifying how a solution might be achieved.
Causal Diagnosis:	Identifying the causes of the problematic state.
Knowledge Specification:	Stating facts and beliefs pertinent to the problem.
Perspective:	Adopting an appropriate point of view on the situation.
Existing Support Environment:	Description of the existing technological environment that can be applied to support the system to be developed.
Stakeholders:	Organizational units, customers, suppliers, competitors.
Process Level Requirements	
Process Description:	A series of steps or tasks designed to produce a product or service.
Process Knowledge Specification:	Facts, rules, beliefs, algorithms, and decisions required to perform a process.
Difficulties, Constraints:	Factors that may prohibit process completion.
Task Level Requirements	
Task Description:	Identification of the sequence of actions required to complete a task.
Task Knowledge Specification:	Facts, rules, beliefs, assumptions, algorithms, and decisions required to perform a task.
Performance Criteria:	Statement that associates an outcome with specific conditions, actions, and constraints.
Roles and Responsibilities:	Individuals or departments who are charged with performing tasks or steps within tasks.
Justification:	Explanations of why specific actions are or are not to be taken.
Information Level Requirements	
Displayed Information:	Data to be presented to end-users in paper or electronic format.
Interface Design:	Language and formats used in presenting "Displayed Information."
Inputs:	Data that must be entered into the system.
Stored Information:	Data saved by the system.
Objects and Events:	Physical entities and occurrences in the world that are relevant to the system.
Relationships Between Objects and Events:	Description of how one object or event is associated with another object or event.
Data Attributes:	Characteristics of objects and events.
Validation Criteria:	Rules that govern the validity of data.
Computations:	Information created by the system.

60, 62]. For example, if an analyst uses the proposed categories to code responses he has received from users, and only 8 of the 32 generic categories are used, this is an indication that more requirements may need to be elicited. The categories can thus be used as a stopping rule for the collection of requirements. This should improve analysts' requirements determination efforts by providing them with guidelines to follow, resulting in more effective and efficient systems development.

For researchers, the generic requirements categories can be used in several ways. In the present research, the categories are used to code subjects' responses to questions from the elicitation techniques being tested. Future research may focus on the usefulness of the various categories and enhancements that may be appropriate.

Hypotheses

Based on the background material discussed above, several hypotheses were generated. Stated in the null form, they are as follows:

Hypothesis 1: The total number of requirements per subject elicited by the task characteristics prompting technique will not differ from the number elicited by either the semantic technique or the syntactic technique.

Hypothesis 2a: The total number of goal level requirements per subject elicited by the task characteristics prompting technique will not differ from the number elicited by either the semantic technique or the syntactic technique.

Hypothesis 2b: The total number of process level requirements per subject elicited by the task characteristics prompting technique will not differ from the number elicited by either the semantic technique or the syntactic technique.

Hypothesis 2c: The total number of task level requirements per subject elicited by the task characteristics prompting technique will not differ from the number elicited by either the semantic technique or the syntactic technique.

Hypothesis 2d: The total number of information level requirements per subject elicited by the task characteristics prompting technique will not differ from the number elicited by either the semantic technique or the syntactic technique.

Hypothesis 3: The breadth of requirements (i.e., number of different requirements categories) elicited by the task characteristics prompting technique will not differ from the breadth elicited by either the semantic technique or the syntactic technique.

Hypothesis 4: Generic requirements evoked by subjects in the task characteristics group will not be independent of the generic requirements evoked by subjects in either the semantic group or the syntactic group.

Method

THIS SECTION DESCRIBES the experimental design, data collection procedures, and methods used to analyze the data.

Experimental Design

Experimental Task

The experiment utilized a short case describing a grocery company interested in developing an Internet-based food shopping system. The case appears in Figure 2. An online shopping application was chosen because while grocery shopping is a common task (allowing subjects to provide user requirements), online grocery shopping systems were very new at this time. It was believed that because of the novelty of such a system, the articulation of requirements would require subjects to be imaginative and not rely on previously formed beliefs. That is, we wanted a task with which subjects were not already familiar. If the case were not novel, subjects' responses could be based on prior experiences with systems of that type, which might bias the results in an unknown manner.

Only one case was utilized for this research because of the labor-intensiveness of the data collection method employed. Because the task utilized a mainstream business case, the results of the study should be generalizable to other types of systems development tasks.

Subjects

Subjects consisted of 45 nonfaculty employees from two universities. The criteria for selection was that subjects must have used a computer system as part of their job function and have had experience using databases. People with information systems development experience were specifically excluded from the subject pool because of the experimenters' judgment that such people might think in terms of system solutions rather than focus on business requirements. We believed that our subjects would naturally adopt the point of view of shoppers using the proposed system. To cause them to consider the requirements of the grocery store as well, we asked them in the experimental case to assume the role of a grocery store manager. This allowed subjects to consider both of the main perspectives useful for specifying requirements.

The use of 45 subjects exceeds the number used in typical studies using the verbal protocol method of data analysis. In a comprehensive review of processing tracing research, Ford et al. [21] cited 17 verbal protocol studies using fewer than 45 subjects and only two studies using more than 45 (for more recent studies and commentary, see, e.g., [6, 11, 17]).³

A completely randomized design was utilized. Subjects were assigned to the three treatment groups, described next, by using a random number table.

Foodco, a regional supermarket chain with stores in several contiguous states, is facing severe competition. To attempt to increase its customer base and therefore its sales, Foodco has decided to implement an online shopping system whereby customers can do their grocery shopping from home via computer.

As the first step in this process Foodco must develop a set of requirements to guide the design and development of the on-line grocery shopping system. You are a Foodco employee. You have been managing a store for 10 years and are on the company-wide Foodco strategic planning committee. As a store manager, you are responsible for all the store's operations and handle all the day to day problems encountered at the store.

You have been appointed to be the person responsible for defining the requirements for the new online shop at home system. A systems analyst from Foodco's computer department will meet with you to find out about Foodco's store operations and how you envision these operations will be accomplished in the new online computer system. To assist in the determination of requirements, the systems analyst will ask you a series of questions designed to help you articulate what should be included in the new system.

Figure 2. Foodco Requirements Elicitation Task

Experimental Groups

An experiment was designed with three groups. One group, labeled the "task characteristics group," received the set of prompts developed in the present research and shown earlier in Table 3. A second group, labeled the "syntactic group," received the who, what, when, where, why, and how questioning approach shown in Table 4. A third group, labeled the "semantic group," received the operationalized version of the questioning scheme devised by Lauer [31] and shown in Table 5.

Procedure

An interviewer participated one-on-one with subjects in a lab setting. The interviewer was a person blind to the hypotheses of the study. All experimental sessions were tape recorded. The interviewer functioned as the systems analyst in the experiment, and the subjects functioned as users. Subjects were given a set of instructions and followed along as the interviewer read the instructions explaining what the subjects were expected to do and what the interviewer would do during the experimental session. The instructions followed the guidelines established by Carroll et al. [8] for verbal protocol studies. Subjects were then given the task description and asked to read it. The interviewer then conducted the questioning process by asking the questions contained

in the appropriate prompting technique. Subjects were asked to speak aloud as they thought about and responded to each question.

Methods of Analysis

Coding Procedure

All tape recorded sessions were transcribed for analysis and coding. The transcribed protocols were the sole source of data used in the analyses. Before coding is possible, protocols first must be parsed into meaningful units. The present research utilized a "context space" parsing scheme developed by Curley et al. [11], who relied on Reichman-Adar [42]. Context spaces are blocks of utterances in which the speaker is discussing a single subject. After the protocols were parsed, they were coded into the generic requirements categories developed by the researchers as part of the present study. The coding was performed by a person unfamiliar with systems development and blind to the hypotheses of the study, using a scoring method that involved tabulating frequencies of items of interest [17, 55]. An example portion of a subject's parsed protocol and the codes assigned are shown in Table 7. The requirement level and generic requirements shown in Table 7 are from the coding scheme displayed previously in Table 6. As a check on the reliability of the coding, a second person coded a random sample of 20 percent of the subjects' protocols. Interrater reliability between the two coders was 92 percent. The first person's codes were deemed reliable and were used in the analyses discussed below.

Measures

Several measures were utilized to assess the information evoked by subjects. Both quantity and differences in category usage across groups were of interest. The primary measures of interest concern the quantity of information. Quantity of information evoked is important because if requirements are not mentioned by users, that information cannot be incorporated into the system. It is assumed that the relevance and usefulness of domain knowledge for application development are not known a priori by the analyst. Thus, eliciting as much information as possible provides the best opportunity for identifying information most relevant to the proposed system.⁴

Quantity of information was measured in two ways. First, the total number of requirements evoked by subjects was counted, for each major requirement category and as a grand total. Second, a "breadth" of requirements measure was calculated by counting the total number of different requirement categories mentioned by subjects. Breadth is important because it shows how well a technique elicits a range of information from subjects. The more different categories touched upon by subjects, the more system requirements that can theoretically be identified.

Quantity of information elicited is of paramount importance in the present research because the techniques being tested are elicitation techniques. The importance of this elicitation step to systems development is clear [15, 59]: Information not mentioned

Table 7. Sample Parsed and Coded Protocol

Prompt: What would your customers want the system to do?				
Parsed Response	Requirement Level	Generic Requirement	Explanation	
It needs to have all the personal information quickly at your fingertips—name, address,	Information	Stored Information (name)	Example of information that must be stored by the system.	
	Information	Stored Information (address)	Another example of information that must be stored by the system.	
	Process	Process Description (billing)	Identifies the need for a billing process.	
whatever information you need for their billing.				
For instance, when their last payment was,	Information	Stored Information (last payment)	Identifies an information requirement associated with billing.	
whatever balance is on their account.	Information	Stored Information (balance)	Another information requirement associated with billing.	
I would imagine in this particular system that you would want to keep track of whatever you know this person orders every week.	Information	Stored Information (items ordered weekly/regularly)	Another example of information that must be stored by the system.	
You want their database set so their staples are in there, so they can constantly go into the database and enter as little as possible.	Task	Performance Criteria (order entry)	Identifies the customer task of order entry and states task performance criteria, specifying that the customer must be able to enter an order as easily as possible.	
Of course, that won't happen until they become your customer; you have to interview them to see what are the standard things they order every week.	Process	Process Description (customer set-up)	Identifies the process of setting up the customer on the system.	

cannot be used in the systems development process. However, the *use* of the information, for example, in the ultimate system design and implementation, is not addressed in the current research.

In addition to the quantity of information measures, one additional measure was of interest. It was possible that the elicitation techniques elicited different information from the users—that is, that there were qualitative differences in the information elicited. If this were the case, it might suggest that a combination of techniques should be used by analysts. To measure whether different requirements were elicited, the generic requirements categories were ranked within each group according to the number of subjects who mentioned that category. This provided an indication of whether subjects in the three groups were emphasizing *different* information.

Results

THIS SECTION DETAILS RESULTS FROM TESTS OF THE HYPOTHESES noted above. Time spent on the task by members of the three groups was not significantly different ($F_{(2,42)} = 1.91$; $p = 0.16$), indicating that any differences between the groups were not a result of the amount of time spent considering the problem.

Quantity of Requirements Elicited

Data concerning requirements elicited from each group for each general level category in the coding scheme and overall are shown in Table 8. The first variable of interest was the total number of requirements elicited by the three prompting techniques. An analysis of variance (ANOVA) was performed to test for differences between the treatment groups.⁵ The ANOVA showed that the means were significantly different ($F_{(2,42)} = 17.05$; $p < 0.0001$) (for all hypothesis tests, we assumed that $\alpha = 0.05$). To test for differences between groups, the Scheffé procedure for multiple comparisons was performed. The contrasts showed that the task characteristics prompts elicited significantly more requirements than both the semantic prompts and the syntactic prompts. These results support the rejection of null Hypothesis 1.

Hypothesis 2 stated that the task characteristics technique and other techniques would not differ in the number of requirements elicited from each of the general level requirements categories in the coding scheme. To test this notion, four additional analyses of variance were performed. The results showed that there were no differences between the techniques for the total *goal level* requirements elicited ($F_{(2,42)} = 0.64$; $p = 0.53$). Consequently, Hypothesis 2a cannot be rejected. Although unexpected, this result has important implications for the use of directed questions in systems analysis practice. The semantic prompting technique contained 20 prompts, 9 of which directed subjects' attention explicitly to system goals. This technique, however, failed to elicit significantly more goal level requirements than either of the other techniques, neither of which had any prompts that directed subjects specifically to goals. This result supports other research that questions the wisdom of eliciting information by asking people about knowledge directly (e.g., [2]).

Table 8. Quantity of Requirements Elicited: Group Means and Standard Deviations

	Total		Goal		Process		Task		Information	
	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev	Mean	Std Dev
Prompting Group										
Syntactic	30.80	16.25	8.47	5.15	19.87	8.41	1.73	2.14	1.40	2.52
Semantic	40.93	12.06	8.40	3.52	26.27	9.31	3.07	2.52	3.13	2.45
Task Characteristics	66.27	20.69	6.67	5.71	40.40	10.51	4.60	3.81	14.60	6.56

A second test showed that there was a significant difference between the groups for the *process level* requirements elicited ($F_{(2,42)} = 17.77$; $p < 0.0001$). Multiple comparisons revealed that the task characteristics technique elicited more process requirements than either of the other techniques (there was no difference between the semantic and syntactic groups). Thus, null Hypothesis 2b is rejected. For the *task level* requirements, the ANOVA showed differences between the groups to be marginally significant ($F_{(2,42)} = 3.23$; $p = 0.049$). Multiple comparisons showed that the task characteristics group elicited more task level requirements than the syntactic technique, but not the semantic technique. Thus, there is some support for rejecting null Hypothesis 2c. For the *information level* requirements, there was a statistically significant difference between the groups ($F_{(2,42)} = 39.61$; $p < 0.0001$). Contrasts showed that the task characteristics prompting technique elicited more information requirements than both the semantic and syntactic techniques (again, there was no difference between the semantic and syntactic groups). These results support the rejection of null Hypothesis 2d.

A second general measure of the quantity of information elicited by the three prompting techniques concerns the breadth of requirements. Breadth of requirements was measured by counting the number of different generic requirement categories into which subjects' utterances were coded. The mean number of different categories utilized by group was as follows: Task Characteristics: 9.1; Semantic: 8.1; Syntactic: 7.1. An analysis of variance showed that there was no statistical difference between the groups ($F_{(2,42)} = 2.47$; $p = 0.10$). Therefore, Hypothesis 3 cannot be rejected. This result may indicate that there is no advantage for any of the groups in the breadth of information elicited. An alternative explanation for the finding, however, is the precision of the coding scheme. For example, the process level and task level contained only three and five categories of requirements, respectively. This limited number of categories may have been insufficient to measure adequately the breadth of requirements elicited by each technique.

Differing Category Usage

The final hypothesis concerns qualitative differences in information elicited by the three prompting techniques. If requirements from different generic categories are elicited by the three techniques, this will provide evidence that more than one technique should be used in combination to gather requirements. To test the degree of dependence of the information evoked, the categories were first ranked in terms of the number of times they were mentioned by subjects in the three groups. Correlations were then performed between the ranks for the task characteristics technique and syntactic technique, and for the task characteristics technique and semantic technique. Because serious violations of normality were present in these data, Spearman's test of rank correlation was utilized [9].

The Spearman test showed that there was a correlation between the generic requirements categories evoked by subjects in the task characteristics group and the syntactic group ($r_s = 0.77$; $p < 0.01$). The comparison between the task characteristics group and the semantic group showed similar results ($r_s = 0.79$; $p < 0.01$). Therefore, Hy-

pothesis 4 cannot be rejected. There were no significant qualitative differences in the types of requirements elicited by each technique. Consequently, the use of more than one technique is not likely to cause users to evoke requirements from different categories. Because the task characteristics technique elicits a greater amount of information than the other techniques tested (as shown in the tests concerning Hypotheses 1 and 2), the results of the Spearman test implicate its use alone.

The rejection of Hypothesis 4 is not a disappointing result for systems analysis practice, because the use of several techniques for eliciting requirements is not appealing. The systems analyst charged with gathering requirements is faced with cost and time constraints, and in the rapid application development environment facing organizations today, use of several techniques would be costly and time-consuming. The results of this study suggest that using more than one technique would be redundant, and the prescription that the task characteristics technique can be used alone with confidence should be welcomed by practitioners.

Discussion

THIS PAPER FURTHERS OUR UNDERSTANDING of requirements elicitation in several ways. First, we have presented a theoretical model of the requirements elicitation task that can be used as a framework for future research. Previously, no task-level model of the elicitation process had been proposed. Without a clear description of the elicitation task, it is difficult to organize and direct research to improve the process. Practitioners can benefit from the model by gaining a fuller understanding of the task, including both the problems and opportunities inherent in the interaction between user and analyst.

Second, we have developed a theory-based prompting technique for eliciting requirements. The technique is based on the new model of the requirements elicitation task and theoretical knowledge from several domains. We have also provided an empirical test of this technique, the first such test of the effectiveness of a prompting technique in the elicitation of requirements. The data show that the task characteristics technique is generally very effective in eliciting information from subjects.

Third, building on the work of Byrd et al. [7], we have created an enhanced taxonomy of generic requirement categories for organizing information gathered in the analysis stage of system development. These categories provide a taxonomy to guide practitioners in the identification and capture of requirements. The taxonomy can benefit from continued research to refine the categories and further test their usefulness.

It is worth noting that the particular prompts used in the task characteristics technique will likely need to be supplemented in a requirements elicitation effort for a major organizational information system. Our goal in this research was not to create *the* set of elicitation prompts for all systems development projects. Rather, our purpose was to demonstrate a methodology for creating prompts based on a model of the requirements elicitation task and theories from various disciplines. The evidence from this study shows that prompts based on this theoretical analysis are likely to lead to the elicitation of more system requirements from users.

The availability of a theory-driven means for constructing context-independent prompts and the generic requirements categories are important for both practice and research. From a practical perspective, the prompting technique offers guidance to systems analysts regarding the types of questions that may be useful in eliciting information from users. The generic categories provide a means for analysts to organize requirements elicited and offer guidance concerning when an analyst can stop eliciting requirements. From a research perspective, the prompts and categories offer a starting point for developing theory-driven methods of requirements elicitation and capture. Further research is needed to refine the prompts and categories, and perhaps to develop additional prompting schemes based on the same underlying theory.

The present research also contributes to a growing understanding of how requirements determination methods affect the process of systems development. As noted, Lauer [31] and Marakas and Elam [35] have used Lauer et al.'s [32] semantic prompting method to test the quality of data-flow diagrams. Such diagrams represent one step in the movement from requirements as they exist in the world to systems that are developed to support those requirements. These diagrams represent one *use* of information elicited by analysts. The present research provides evidence for the effectiveness of techniques in terms of the *elicitation* of information from users. The elicitation step provides the raw material for system diagrams, and therefore is in some sense more fundamental. Hence, the present research fills another gap in our knowledge of systems development in general and requirements elicitation in particular. Future research extending the present findings should provide further understanding of the crucial role of requirements elicitation in the systems development process.

Acknowledgments: The authors thank Mitzi Pitts, Ramesh Venkataraman, John Durrett, W.J. Conover, the Editor-in-chief, and three anonymous reviewers for their helpful comments on previous versions of this paper. This research was supported in part by a grant from the David D. Lattanze Center for Executive Studies in Information Systems.

NOTES

1. Moody, Blanton, and Cheney [37] tested two types of interviewing techniques for requirements determination, but their focus was not on prompting strategies. Thus, their study was at a higher level of abstraction than the current research.

2. In operationalizing the interrogatories technique and, in particular, the semantic questioning scheme [32], care was taken to represent the spirit of the original questions as faithfully as possible in the prompts constructed. It is of course possible that different operationalizations of the interrogatory and the semantic questioning techniques would have yielded different experimental results.

3. Although the number of subjects exceeded the number typically used in verbal protocol analysis studies, it was important to know whether 15 subjects per group was large enough to capture meaningful differences between groups. We were unable to estimate the number of subjects needed *a priori* because we did not have strong expectations about the effect size (that is, the differences in group means expressed in units of standard deviation). We performed a post hoc power analysis to determine the power of the tests used given the $\alpha = 0.05$ significance criterion and the means and standard deviations we found for each dependent variable [52]. The sample size of 15 subjects per group was adequate to identify potential differences

between groups of at least one standard deviation (which is quite conservative, since one standard deviation is larger than almost all effect sizes found in behavioral sciences data [52]) with a probability of 0.99 for all measures except the goal level requirements. For the goal level requirements, the observed difference was less than one standard deviation and the power of the test was 0.58. Thus, the sample size was more than adequate in general given the effect sizes in these variables.

4. The availability of evoked information is arguably more important in requirements determination for systems development than in other domains. In decision making, for example, the availability of more information does not necessarily lead to better decisions [61]. In domains such as decision making, large amounts of information must ultimately be reduced to a limited number of judgments. In requirements determination, the same funneling effect is not present. Each element of information can conceivably become a requirement for the proposed system. Thus, one measure of the *quality* of a requirements elicitation technique is arguably the quantity of requirements it elicits.

5. In several of the tests, analyses revealed violations of the normality assumption underlying the analysis of variance procedure. In all cases, normal probability plots were examined and the departures from normality were not deemed serious. Because the analysis of variance procedure is robust regarding minor violations of normality [52], ANOVA results were utilized in the analyses unless otherwise noted.

REFERENCES

1. Agarwal, R., and Tanniru, M. Knowledge acquisition using structured interviewing: an empirical investigation. *Journal of Management Information Systems*, 7, 1 (Summer 1990), 123–140.
2. Benson, P.G.; Curley, S.P.; and Smith, G.F. Belief assessment: an underdeveloped phase of probability elicitation. *Management Science*, 41, 10 (1995), 1639–1653.
3. Bostrom, R.P. Successful applications of communications techniques to improve the systems development process. *Information & Management*, 16, 5 (1989), 279–295.
4. Brody, R. *Problem Solving*. New York: Human Sciences Press, 1982.
5. Brooks, F.P. No silver bullet: essence and accidents of software engineering. *Computer*, 20, 4 (1987), 10–19.
6. Browne, G.J.; Curley, S.P.; and Benson, P.G. Evoking information in probability assessment: knowledge maps and reasoning-based directed questions. *Management Science*, 43, 1 (1997), 1–14.
7. Byrd, T.A.; Cossick, K.L.; and Zmud, R.W. A synthesis of research on requirements analysis and knowledge acquisition. *MIS Quarterly*, 16, 1 (1992), 117–138.
8. Carroll, J.S.; Bazerman, M.H.; and Maury, R. Negotiator cognitions: a descriptive approach to negotiators' understanding of their opponents. *Organizational Behavior and Human Decision Processes*, 41, 3 (1988), 352–370.
9. Conover, W.J. *Practical Non-Parametric Statistics*. New York: John Wiley & Sons, 1980.
10. Couger, J.D. *Creativity and Innovation in Information Systems Organizations*. Danvers, MA: Boyd and Fraser, 1996.
11. Curley, S.P.; Browne, G.J.; Smith, G.F.; and Benson, P.G. Arguments in the practical reasoning underlying constructed probability responses. *Journal of Behavioral Decision Making*, 8, 1 (1995), 1–20.
12. Dalal, N.P., and Yadav, S.B. The design of a knowledge-based decision support system to support the information analyst in determining requirements. *Decision Sciences*, 23, 6 (1992), 1373–1388.
13. D'Andrade, R.G., and Wish, M. Speech act theory in quantitative research on interpersonal behavior. *Discourse Processes*, 8, 2 (1985), 229–259.
14. Davis, A.M. A comparison of techniques for the specification of external system behavior. *Communications of the ACM*, 31, 9 (1988), 1098–1115.
15. Davis, G.B. Strategies for information requirements determination. *IBM Systems Journal*, 21, 1 (1982), 4–30.

16. Dery, D. *Problem Definition in Policy Analysis*. Lawrence: University of Kansas Press, 1984.
17. Ericsson, K.A., and Simon, H.A. *Protocol Analysis: Verbal Reports as Data*. Cambridge, MA: MIT Press, 1984.
18. Ewusi-Mensah, K. Critical issues in abandoned information systems projects. *Communications of the ACM*, 40, 9 (1997), 74–80.
19. Fischhoff, B. Eliciting knowledge for analytical representations. *IEEE Transactions on Systems, Man, and Cybernetics*, 19, 3 (1989), 448–461.
20. Fischhoff, B., and Bar-Hillel, M. Focusing techniques: a shortcut to improving probability judgments? *Organizational Behavior and Human Decision Processes*, 34, 2 (1984), 175–194.
21. Ford, J.K.; Schmitt, N.; Schechtman, S.L.; Hults, B.M.; and Doherty, M.L. Process tracing methods: contributions, problems, and neglected research questions. *Organizational Behavior and Human Decision Processes*, 43, 1 (1989), 75–117.
22. Goel, V., and Pirolli, P. Motivating the notion of generic design within information-processing theory: the design problem space. *AI Magazine*, 10, 1 (1989), 18–36.
23. Graesser, A.C.; Person, N.; and Huber, J. Mechanisms that generate questions. In T.W. Lauer, E. Peacock, and A.C. Graesser (eds.), *Questions and Information Systems*. Hillsdale, NJ: Lawrence Erlbaum, 1992, pp. 167–187.
24. Guinan, P.J.; Coopridge, J.G.; and Faraj, S. Enabling software development team performance during requirements definition: a behavioral versus technical approach. *Information Systems Research*, 9, 2 (1998), 101–125.
25. Hammer, M., and Champy, J. *Reengineering the Corporation: A Manifesto for Business Revolution*. New York: HarperCollins, 1993.
26. Kahneman, D., and Tversky, A. The simulation heuristic. In D. Kahneman, P. Slovic, and A. Tversky (eds.), *Judgment Under Uncertainty: Heuristics and Biases*. Cambridge: Cambridge University Press, 1982, pp. 201–208.
27. Kendall, K.E., and Kendall, J.E. *Systems Analysis and Design*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
28. Kepner, C.H., and Tregoe, B.B. *The New Rational Manager*. Princeton, NJ: Princeton University Press, 1981.
29. Koriati, A.; Lichtenstein, S.; and Fischhoff, B. Reasons for confidence. *Journal of Experimental Psychology: Human Learning and Memory*, 6, 2 (1980), 107–118.
30. Larsen, T.J., and Naumann, J.D. An experimental comparison of abstract and concrete representations in systems analysis. *Information & Management*, 22, 1 (1992), 29–40.
31. Lauer, T.W. Development of a generic inquiry based problem analysis method. Working paper, David D. Lattanze Center for Executive Studies in Information Systems, Loyola College, Baltimore, MD, 1994.
32. Lauer, T.W.; Peacock, E.; and Jacobs, S.M. Question generation and the systems analysis process. In T.W. Lauer, E. Peacock, and A.C. Graesser (eds.), *Questions and Information Systems*. Hillsdale, NJ: Lawrence Erlbaum, 1992, pp. 47–61.
33. Lehnert, W.G. *The Process of Question Answering*. Hillsdale, NJ: Lawrence Erlbaum, 1978.
34. Leifer, R.; Lee, S.; and Durgue, J. Deep structures: real information requirements determination. *Information & Management*, 27, 5 (1994), 275–285.
35. Marakas, G.M., and Elam, J.J. Semantic structuring in analyst acquisition and representation of facts in requirements analysis. *Information Systems Research*, 9, 1 (1998), 37–63.
36. McMaster, M., and Grinder, J. *Precision: A New Approach to Communication*. Beverly Hills, CA: Precision Models, 1980.
37. Moody, J.W.; Blanton, J.E.; and Cheney, P.H. A theoretically grounded approach to assist memory recall during information requirements determination. *Journal of Management Information Systems*, 15, 1 (1998), 79–98.
38. Newell, A., and Simon, H.A. *Human Problem Solving*. Englewood Cliffs, NJ: Prentice Hall, 1972.
39. Perkins, D.N.; Allen, R.; and Hafner, J. Difficulties in everyday reasoning. In W. Maxwell (ed.), *Thinking: The Expanding Frontier*. Philadelphia: Franklin Institute Press, 1983, pp. 177–189.
40. Pitts, M.G. The Use of Evaluative Stopping Rules in Information Requirements Determination: An Empirical Investigation of Systems Analyst Behavior. Ph.D. dissertation, University of Maryland, 1999.

41. Potts, C.; Takahashi, K.; and Anton, A.I. Inquiry-based requirements analysis. *IEEE Software*, 11, 2 (1994), 21–32.
42. Reichman-Adar, R. Extended person-machine interface. *Artificial Intelligence*, 22, 2 (1984), 157–218.
43. Reisberg, D. *Cognition*. New York: W.W. Norton & Company, 1997.
44. Robillard, P.N. The role of knowledge in software development. *Communications of the ACM*, 42, 1 (1999), 87–92.
45. Rummler, G.A., and Brache, A.P. *Improving Performance: How to Manage the White Space on the Organization Chart*. San Francisco: Jossey-Bass, 1990.
46. Simon, H.A. *The Sciences of the Artificial*. Cambridge: Cambridge University Press, 1981.
47. Smith, G.F. *Quality Problem Solving*. Milwaukee: ASQ Quality Press, 1998.
48. Smith, G.F. Towards a heuristic theory of problem structuring. *Management Science*, 34, 12 (1988), 1489–1506.
49. Smith, G.F., and Browne, G.J. Conceptual foundations of design problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 23, 5 (1993), 1209–1219.
50. Stacy, W., and Macmillan, J. Cognitive bias in software engineering. *Communications of the ACM*, 38, 6 (1995), 57–63.
51. Standish Group. Chaos. Research paper, The Standish Group International, Inc., 1996.
52. Stevens, J. *Applied Multivariate Statistics for the Social Sciences*. Mahwah, NJ: Erlbaum, 1996.
53. Teng, J.T.C., and Sethi, V. A comparison of information requirements analysis methods: an experimental study. *Data Base*, 20, 4 (1990), 27–39.
54. Thomas, J.C., and Carroll, J.M. Human factors in communications. *IBM Systems Journal*, 20, 2 (1981), 237–263.
55. Todd, P., and Benbasat, I. Process tracing methods in decision support systems research: exploring the black box. *MIS Quarterly*, 11, 4 (1987), 493–512.
56. Toulmin, S.E. *The Uses of Argument*. Cambridge: Cambridge University Press, 1958.
57. Turner, J.A. A comparison of the process of knowledge elicitation with that of information requirements determination. In W.W. Cotterman, and J.A. Senn (eds.), *Challenges and Strategies for Research in Systems Development*. New York: John Wiley & Sons, 1992, pp. 415–430.
58. Valusek, J.R., and Fryback, D.G. Information requirements determination: obstacles within, among and between participants. In R. Galliers (ed.), *Information Analysis: Selected Readings*. Reading, MA: Addison Wesley, 1987, pp. 139–151.
59. Vessey, I., and Conger, S. Requirements specification: learning object, process, and data methodologies. *Communications of the ACM*, 37, 5 (1994), 102–113.
60. Vitalari, N.P. Structuring the requirements analysis process for information systems: a propositional viewpoint. In W.W. Cotterman, and J.A. Senn (eds.), *Challenges and Strategies for Research in Systems Development*. New York: John Wiley & Sons, 1992, pp. 163–179.
61. von Winterfeldt, D., and Edwards, W. *Decision Analysis and Behavioral Research*. Cambridge: Cambridge University Press, 1986.
62. Watson, H.J., and Frolick, M.N. Determining information requirements for an EIS. *MIS Quarterly*, 17, 3 (1993), 255–269.
63. Wetherbe, J.C. Executive information requirements: getting it right. *MIS Quarterly*, 15, 1 (1991), 51–65.
64. Whitten, J.L., and Bentley, L.D. *Systems Analysis and Design Methods*. Burr Ridge, IL: Irwin, 1998.
65. Yadav, S.B.; Bravoco, R.R.; Chatfield, A.T.; and Rajkumar, T.M. Comparison of analysis techniques for information requirement determination. *Communications of the ACM*, 31, 9 (1988), 1090–1097.
66. Zmud, R.W. *Information Systems in Organizations*. Glenview, IL: Scott-Foresman, 1983.
67. Zmud, R.W.; Anthony, W.P.; and Stair, R. The use of mental imagery as a requirements analysis technique. In W.W. Cotterman, and J.A. Senn (eds.), *Challenges and Strategies for Research in Systems Development*. New York: John Wiley & Sons, 1992, pp. 337–350.