A Unified Model of Requirements Elicitation

ANN M. HICKEY AND ALAN M. DAVIS

ANN M. HICKEY is an Assistant Professor of Information Systems at the University of Colorado at Colorado Springs. She previously worked for 17 years as a program manager and senior systems analyst for the Department of Defense. Her research focuses on scenarios, collaborative requirements elicitation, and elicitation technique selection and has been published in the Journal of Management Information Systems, Database for Advances in Information Systems, Requirements Engineering Journal, and in national and international conference proceedings. She received her B.A. from Dartmouth College and her M.S. and Ph.D. in MIS from the University of Arizona.

ALAN M. DAVIS is a Professor of Information Systems at the University of Colorado at Colorado Springs. He has consulted for many corporations over the past 25 years, including Boeing, Cigna Insurance, FedEx, Fujitsu, General Electric, IBM, Loral, McDonald’s, MCI, Mitsubishi Electric, Rockwell, Schlumberger, and Storage Tek. Previously, he was chairman and CEO of Omni-Vista, Inc., a Vice President at BTG, Inc., a Director of R&D at GTE Communication Systems, and Director of the Software Technology Center at GTE Laboratories. He was Editor-in-Chief of IEEE Software from 1994 to 1998. He is an associate editor for the Journal of Systems and Software (1987–present) and was an editor for Communications of the ACM (1981–1991). He is the author of Software Requirements: Objects, Functions and States (Prentice Hall, 1990; 1993), 201 Principles of Software Development (McGraw-Hill, 1995), and the forthcoming Just Enough Requirements Management (Dorset House, 2004). He is the founder of the IEEE International Conferences of Requirements Engineering, and served as general chair of its first conference in 1994. He has been a fellow of the IEEE since 1994. He earned his M.S. and Ph.D. in Computer Science from the University of Illinois.

ABSTRACT: Effective requirements elicitation is essential to the success of software development projects. Many papers have been written that promulgate specific elicitation methods. A few model elicitation in general. However, none have yet modeled elicitation in a way that makes clear the critical role played by situational knowledge. This paper presents a unified model of the requirements elicitation process that emphasizes the iterative nature of elicitation as it transforms the current state of the requirements and the situation to an improved understanding of the requirements and, potentially, a modified situation. One meta-process of requirements elicitation, selection of an appropriate elicitation technique, is also captured in the model. The values of this model are: (1) an improved understanding of elicitation helps analysts improve their elicitation efforts and (2) as we improve our ability to perform elicitation, we improve the likelihood that systems we create will meet their intended customers’ needs.

KEY WORDS AND PHRASES: requirements analysis, requirements elicitation, systems analysis.
Requirements elicitation is recognized as one of the most critical activities of software development [2]: poor execution of elicitation will almost guarantee that the final project is a complete failure. Since software project failures are so rampant [43], it is quite likely that improving how the industry performs elicitation could have a dramatic effect on the success record of the industry [21]. Improving requirements elicitation requires us to first understand it. Although many papers have been written that define elicitation, or prescribe a specific technique to perform during elicitation, none have yet defined a unified model of the elicitation process that emphasizes the knowledge needed to effectively elicit requirements from stakeholders.

To better understand the importance of the current paper, let us contrast its goal with the goal of the many dozens of writings (e.g., [11, 24, 46]) that present a specific methodology for elicitation broken down into multiple steps. Studying the steps provides the reader with an understanding of one particular way of doing elicitation. Some writings (e.g., [4, 5, 13, 28, 29, 30, 32, 34]) provide limited insight into when a methodology or specific elicitation technique may or may not be applicable. This paper provides a model for elicitation in general. It then extends that model to include a model of the elicitation technique selection process. Better understanding of elicitation and the factors that should be considered during elicitation technique selection will improve the quality of the requirements elicitation process and, ultimately, increase the success of software development projects [17].

Overview of the Research Domain

Software development is the activity of creating a software system that, when used, solves some users’ problems, leverages their opportunities, or satisfies their needs. Classic software development follows a well-defined series of phases, traditionally called a waterfall model [38]. More recently, software development is performed iteratively, where development activities are repeated multiple times, resulting in a time series of successively more sophisticated products. In the former case, requirements activities are performed ostensibly at the beginning of the software development process. However, with the inevitable onslaught of constantly changing needs, requirements activities need to be performed regularly. In the latter case, requirements activities are performed ostensibly at the beginning of each iteration. As in the former case, requirements change constantly. If the iterations are close enough together, it is usually easier to defer requirements changes to the beginning of a subsequent iteration and thus little time need be expended within any iteration performing additional requirements activities. Regardless of when performed, the requirements activities are essential to understanding users’ needs and, therefore, to the success of the software development effort.

Requirements Activities

The requirements process is often described as a series of activities such as:1
• **Elicitation**—learning, uncovering, extracting, surfacing, or discovering needs of customers, users, and other potential stakeholders.

• **Analysis**—analyzing the information elicited from stakeholders to generate a list of candidate requirements, often by creating and analyzing models of requirements, with the goals of increasing understanding and searching for incompleteness and inconsistency.

• **Triage**—determining which subset of the requirements ascertained by elicitation and analysis is appropriate to be addressed in specific releases of a system.

• **Specification**—documenting the desired external behavior of a system.

• **Verification**—determining the reasonableness, consistency, completeness, suitability, and lack of defects in a set of requirements.

The majority of existing models of the requirements process show it as an ordered sequence of activities. In reality, requirements activities are not performed sequentially, but iteratively and in parallel, as shown conceptually in Figure 1.2 Other models show how information flows among the activities [31], emphasizing the inputs and outputs of each activity (e.g., [28]). Most requirements models include a requirements elicitation activity, either as a separate activity or as part of another requirements activity.

**Requirements Elicitation**

As mentioned earlier, elicitation is all about determining needs of stakeholders. Most models of requirements elicitation focus on specific methodologies or techniques. For example, the Robertsons’ Volere requirements methodology includes a detailed model of its requirements elicitation activities with inputs, outputs, and recommended techniques for each activity [37]. Several researchers [22, 23] have developed specific models that define how to use scenarios for requirements elicitation. Sutcliffe
and Ryan [44] present a model of elicitation that combines scenarios, prototypes, and design rationale. Sommerville et al. [42] describe their approach for using viewpoints to elicit requirements. Some of the most detailed elicitation models describe collaborative requirements workshops such as Joint Application Development (JAD) [46]. For example, Gottesdiener’s [14] discussion of requirements workshops (1) presents models of the inputs/outputs; (2) describes how various requirements models can be used to answer the “six great focus questions” (who, what, why, when, where, how); and (3) provides several sample workshop process examples.

Very few general models of elicitation exist [3]. Some authors provide overall principles for elicitation [10, 36]. Others describe general approaches (e.g., top-down versus bottom-up) [9, 14]. Maciaszek [33] describes the influences during requirements elicitation by showing how analysts, domain experts, and customers interact to provide domain knowledge and use case requirements that are used to produce business class and use case models. Dean et al. [7] also take a model-centric view in their process model of collaborative requirements elicitation and validation (CREV), which defines how activity, data, and scenario models can be used together with prototypes to determine requirements. They also define a model of user communication, showing the levels of user participation (from a single user representative to the entire user community) for different elicitation techniques [7]. Browne and Rogich [3] also look at a communication-based model of elicitation, but focus on the cognitive aspects of user/analyst interaction. However, no one has defined a general model of elicitation that emphasizes the knowledge required to effectively elicit requirements.

Requirements Elicitation Technique Selection

Many requirements elicitation methodologies and techniques exist, all with the common aim to assist analysts in understanding needs [32]. Although some analysts think that just one methodology, or just one technique, is applicable to all situations, one methodology or technique cannot possibly be sufficient for all conditions [6, 12, 14, 28, 32, 34, 47].

Analysts select a particular elicitation technique for any combination of four reasons: (1) it is the only technique that the analyst knows; (2) it is the analyst’s favorite technique for all situations; (3) the analyst is following some explicit methodology, and that methodology prescribes a particular technique at the particular time; and (4) the analyst understands intuitively that the technique is effective in the current circumstances. Clearly, the fourth reason demonstrates the most sophistication by the analyst. We hypothesize that such maturity leads to improved understanding of stakeholders’ needs, and thus a higher likelihood that a resulting system will satisfy those needs. Unfortunately, novice and expert systems analysts differ significantly in ability and maturity [40]. Most practicing analysts simply do not have the insight necessary to make such an informed decision, and therefore rely on one of the first three reasons. Although some limited guidance has been provided on when various elicitation techniques should be used (see, e.g., [4, 5, 13, 28, 29, 30, 32, 34]), no one has
defined a general model of the elicitation technique selection process and the factors that should be considered when selecting techniques.

A New Unified Model of Requirements Elicitation

In order to improve our knowledge of the elicitation process, elicitation methodologies and techniques, and the elicitation technique selection process, we propose a new unified model of elicitation. In previous sections, we defined the scope of elicitation, and presented other researchers’ models of elicitation. As we saw, two classes of models have been documented: (1) those that capture a specific methodology or technique and (2) those that model elicitation in general. In the first class, the models possess three major weaknesses:

1. Each describes a specific elicitation methodology or technique.
2. Each prescribes a specific series of steps, each with its own predefined technique. In effect, they are saying “one size [methodology] fits all.”
3. Each fails to model either the technique selection process or the situational characteristics that drive that decision process.

In the second class, the models possess different weaknesses:

4. Most have underlying, but unstated, assumptions. One noteworthy exception is Brown and Rogich [3].
5. None discuss the role of knowledge in performing elicitation or in selecting elicitation techniques. This knowledge includes (1) the current problem, solution, and project characteristics; (2) the awareness of which requirements are known and which are still to be determined; and (3) knowledge of the relationship of the current problem, solution, and project characteristics and the state of the requirements to the selection of an elicitation technique.

To overcome these weaknesses, we first developed a conceptual model that builds on Loucopoulos and Karakostas’s [31] model of requirements elicitation by adding an elicitation technique selection process and emphasizing the knowledge driving each of those processes (see Figure 2). Note that the elicitation technique selection process is driven by problem, solution, and project domain characteristics as well as the state of the requirements. The domain characteristics are included as drivers based on the increasing recognition that “one size does not fit all” and that techniques must be chosen based on the specific situation [6, 12, 14, 28, 32, 34, 47]. The state of the requirements drive technique selection because techniques vary significantly in their ability to elicit different types of requirements [4, 5, 14, 29]. For example, data modeling techniques are great for eliciting data requirements, but are of no use when trying to identify usability requirements. Therefore, we must know the type of requirements we are seeking to choose the best technique. In the next section, we formalize this model and use the new model to explore these relationships in much greater depth.
The Model

This section of the paper describes a formal model of elicitation that addresses the weaknesses in existing elicitation models and represents a generalization of all known elicitation methodologies and techniques. Its goals are to:

- explicitly highlight the role knowledge plays in performing both elicitation and elicitation technique selection;
- provide a unified framework for understanding the role of requirements elicitation in software development;
- identify the factors analysts should consider when selecting an elicitation technique;
- describe how any elicitation methodology could be represented in terms of that model;
- show what assumptions existing elicitation methodologies make about the situation;
- identify how easily one can tailor existing methodologies for unique situations; and
- show how one can create new elicitation methodologies easily by defining situational characteristics and then observing and recording the resultant instances of methodologies.

On any project, an analyst performing elicitation moves through a series of activities. The purpose of each activity is to bring the parties closer to a common under-
standing of the requirements they wish to address; \( R_1, R_2, R_3 \), and so on. The series of activities can be viewed as the application of a series of mathematical functions, \( elicit_1, elicit_2, \ldots \), each of which creates new requirements by applying an elicitation technique. Thus, each step of elicitation can be defined as

\[
elicit_i(R_i, S_i, t_i) \rightarrow R_{i+1}, S_{i+1}.
\]

That is, at step \( i \) of the elicitation process, elicitation applies technique \( t_i \) when situation \( S_i \) exists and \( R_i \) captures the current state of knowledge of the requirements. The application of technique \( t_i \) results in a new, and hopefully improved, state of the requirements \( R_{i+1} \), and may also change the situation to a new situation \( S_{i+1} \), as shown in Figure 3(a). Note that \( t_i \in T \), the set of all known elicitation techniques. The sets \( R, S \), and \( T \) are further described in the following three sections.

Clearly this slice of the model addresses problems 1 and 2 above; the methodology and the techniques applied at each step are not predefined, but, instead, the elicitation technique at each step should be selected because it is the most applicable to the current situation or is the most likely to improve the state of the requirements.

To address problem 3, we need to look at the elicitation technique selection process more closely. As shown in Figure 2, selection of an elicitation technique should consider:

- What requirements are known and what are not yet known. These are likely to change dynamically throughout the life of the project. This is represented in our model as \( R_i \).
- The current situation, represented in our model as \( S_i \). Specific situational characteristics include:
  - Characteristics of the problem domain. These are usually static throughout the life of a project.
  - Characteristics of the solution domain. These are likely to change whenever a new type of solution to the problem is proposed.
  - Characteristics of the project. These are likely to change whenever culture or management changes. Certain project characteristics may also change during requirements elicitation (e.g., use of a specific elicitation technique may change the level of trust and cooperation between developers and users).

Therefore, as shown in Figure 3(b), elicitation technique selection can be modeled as a selector function:

\[
\sigma (R_i, S_i, \chi(T)) \rightarrow \{ T_i \subseteq T | \text{if } t \in T_i \text{ then } t \text{ is applicable in situation } S_i \text{ when the current state of the requirements is } R_i \},
\]

given the set of all elicitation techniques \( T \), and their characteristics \( \chi(T) \). These characteristics capture the inherent aspects of elicitation techniques, such as whether they aid in reducing ambiguity, whether they are effective at helping people converge on a solution, whether they help resolve conflict, whether they help to raise new issues, and so on. They are static and identical for all projects. The goal of the selector function \( \sigma \) is to identify the best possible match between the characteristics of the tech-
niques and the current state of the requirements and situation. For example, if the requirements are unclear, techniques that reduce ambiguity may be helpful.

Since the *elicit* function requires just one elicitation technique and the above selector function creates a set of applicable techniques $T_i$, we must also define a personal selector function,

$$\pi(T_i, P) \rightarrow t_i \in T_i,$$

where analysts apply their own personal preferences, $P$, to select just one technique from the set of applicable elicitation techniques. In some cases, an analyst may have a simplified personal selector function $\Pi$ that always results in the same preferred technique $t$ (i.e., $\Pi(P) \rightarrow t$) that he or she uses repeatedly, and only varies from that route when it is impossible to use. In such cases, requirements elicitation at step $i$ becomes

$$elicit(R_i, S_i, t) \rightarrow R_{i+1}, S_{i+1},$$

which is appropriate as long as $t \in \sigma (R_i, S_i, \chi(T))$.

Using the more general personal selector function $\pi$, requirements elicitation at step $i$ thus becomes

$$elicit(R_i, S_i, \pi(\sigma(R_i, S_i, \chi(T)), P)) \rightarrow R_{i+1}, S_{i+1}.$$  

Note that this function models the combination of the elicitation and technique selection processes in Figure 2. This combined function addresses problems 4 and 5 of other elicitation models by clearly identifying the detailed assumptions and knowledge required for each element of the model. Finally, on any project, elicitation would be performed multiple times until the team is satisfied with the requirements that have been thus far gathered, as shown in Figure 4.

**Description of $R_i$**

A requirement is any externally observable characteristic of a desired system. However, our definition of $R_i$ is not simply the set of these specific requirements known at the beginning of step $i$. It is broader in two ways.
First, it includes not only requirements, but also needs, problems, wants, desires, business rules, features, goals, and the like (see the Volere template [37] for additional items). Although many view requirements as the only output of the elicitation process, in fact these broader types of requirements information are essential to success for several reasons:

- Information about needs, problems, wants, and so on is what drives discovery of the specific requirements.
- Cross-checking this broader information against the requirements the analysts already know helps analysts determine what types of requirements they still need to know. “Knowing what you do not know” is one of the biggest challenges faced by analysts, but is necessary if the analyst is to elicit that information effectively (or at all)!
- This broader requirements information is a mandatory part of most requirements specifications (e.g., [37, 45]). It provides the background information and context necessary to understand the requirements.

Second, $R$ also captures the degree of understanding concerning each of these entities. Elements within any of these entities, such as, a specific goal, or a specific need, or a specific requirement, can exist anywhere along the spectrum from fully understood to fuzzy to completely unknown. Analysts must understand whether a requirement is fully understood or whether additional clarification is necessary so they can select techniques that aid in clarification of the specific type of requirement. Again, this is a function of “knowing what you do not know.”

**Description of $S_i$**

$S_i$ represents all those characteristics of the situation that influence elicitation or elicitation technique selection. As shown in Figure 2, these situational characteristics fall into the following three broad categories:

- **Characteristics of the problem domain.** This includes inherent characteristics of the problem, such as how well the problem is understood, the complexity of the...
problem, the type of application domain (e.g., business versus engineering), the importance of specific nonfunctional requirements, and whether there is an existing manual or automated system.

- **Characteristics of the solution domain.** This includes characteristics of the solution such as the type of solution anticipated (e.g., application versus system versus embedded software) and whether the solution will be developed in-house, through outsourced custom development, or by purchasing commercial off-the-shelf software (COTS).
- **Characteristics of the project domain.** This includes inherent characteristics of the organization, project, and all the individuals involved in the project. More specifically, this includes:
  - Inherent characteristics of the organization and project, such as whether a specific methodology must be used, level of resources or time constraints, or degree of personnel turnover, which may necessitate additional documentation to ensure continuity.
  - Characteristics of the customers and users who define the requirements such as the number of different users, the diversity of needs, level of experience, willingness to work together to reach consensus, and ability to articulate requirements.
  - Important characteristics of developers who use the customers’ requirements to build the system relating to their experience with the problem and solution domain as well as with software development.
  - Key characteristics of analysts who will facilitate the elicitation process, especially their leadership, communication, analytical, and technical knowledge, skills, and abilities.

Using situational characteristics specified for other purposes (e.g., [1, 26, 27, 39]), we have identified over 50 situational characteristics that could influence elicitation technique selection. We have compiled these characteristics into a situational ontology [19] and are currently interviewing expert analysts to verify which of these factors are the primary drivers for technique selection [20].

**Description of \( T_i \)**

An elicitation technique, \( t \), is a documented series of steps, along with rules for their performance, to assist analysts in the elicitation of requirements. Techniques are often combined together into a methodology, which defines specific activities and techniques that an analyst should perform to achieve some goal. We use the term **technique** in its broadest sense, that is, it **must** include a description of what to do, and it **could** include ways of doing it, tools to use in doing it, and notations to use while doing it. And, we include any technique that helps to elicit requirements from stakeholders—even if the technique is traditionally viewed as supporting a different requirements activity. Interviewing, questionnaires, observation, modeling, prototyping, and collaborative requirements workshops are just a few of the many hundreds of elicitation
techniques available today. To better understand this vast array of techniques and how they are similar/differ, we have also created a technique ontology that characterizes techniques based on physical and temporal co-location requirements, record-keeping responsibility, analyst role, support for convergence/divergence, anonymity, number of stakeholders supported, human versus product focus, direct versus indirect (e.g., team-building) support of elicitation, and whether it is tool-based [19].

We can then match these technique characteristics ($\chi(T)$) to the situation and the state of the requirements to identify appropriate elicitation techniques. The result of this mapping is $T_i$, the set of techniques that are applicable at any one step $i$ of the elicitation process. These techniques are the ones that make sense given the state of the situation $S_i$ and that can aid in moving the state of the requirements $R_i$ toward an improved state of knowledge about the requirements $R_{i+1}$.

How to Use the Model

The description of the model itself demonstrates how the model can be used to achieve its first three goals. The purpose of this section is to describe how the model can be used to achieve the remaining goals relating to elicitation methodologies.

A requirements elicitation methodology defines a series of steps that an analyst should perform. Every elicitation methodology, $M_j$, containing $n$ steps can be characterized as

$$M_j = \text{elicit}_1, \text{elicit}_2, \ldots, \text{elicit}_n,$$

and the state of the requirements (i.e., those that have been uncovered and those that have not been uncovered) as a result of applying all $n$ steps of methodology $M_j$ are

$$R_e = \text{elicit}_1(\ldots(\text{elicit}_1(\ldots(\text{elicit}_1(R_1, S_1), t_{j1}), t_{j2}), \ldots), t_{jn}),$$

where $R_i$ is the state of the requirements at the beginning of the project, $S_i$ is the situation at the beginning of the project, and $t_{j1}, t_{j2}, \ldots, t_{jn}$ are the $n$ steps prescribed by methodology $M_j$.

Notice how the function highlights the assumptions that every methodology makes. When the methodology states that the analyst should perform some technique $t_i$ at step $i$, the methodology is making the assumptions that $S_i$ is true and that the requirements state is $R_i$. But given all the variation among people and problems, how could such an assumption possibly be made a priori?

Figure 5 shows us what happens as step $i$ of a methodology is about to begin. Notice that our model of elicitation makes clear the existence of a set of techniques $T_i$ applicable given the current situation and state of requirements. Notice further than any $t \in T_i$ is applicable.

But most methodologies prescribe a specific next step using a specific technique. If that technique is among the members of $T_i$ as shown in Figure 6(a), then the methodology has provided useful, although perhaps excessively restrictive, guidance. On the other hand, if that technique is not among the members of $T_i$ as shown in Figure 6(b), then the methodology is actually providing the analyst with poor direction.
To tailor an existing methodology so that it makes sense, follow this simple procedure at each step \( i \):

1. Examine the state of the requirements, \( R_i \), including what requirements are known and what requirements still need to be discovered.
2. Examine the characteristics of the problem, the solution, and the project, \( S_i \).
3. Determine if the technique \( t_i \) being suggested by the methodology is a member of \( \sigma(R_i, S_i, \chi(T)) \), as shown in Figure 6(a). If so, you should proceed with the application of \( t_i \) as prescribed by the methodology.
4. If the technique \( t_i \) being suggested by the methodology is not a member of \( \sigma(R_i, S_i, \chi(T)) \), as shown in Figure 6(b), then select an alternative technique, that is, \( \pi(\sigma(R_i, S_i, \chi(T)), P) \), and proceed with its use in lieu of \( t_i \).
To create a new methodology for your unique situation, or if you do not want to “follow a methodology” but just want to do elicitation in a way that makes most sense, follow this simple procedure at each step $i$:

1. Examine the state of the known requirements, $R_i$.
2. Examine the characteristics of the problem, the solution, and the project, $S_i$.
3. Using your personal preferences, $P$, select a technique $t_i$ out of the set of all applicable techniques, that is, apply the function $\pi(\sigma(R_i, S_i, \chi(T)), P))$.

**Evaluating the Model**

THREE OF THE KEY GOALS OF THIS UNIFIED MODEL of requirement elicitation were to be able to use the model to (1) analyze existing methodologies, (2) represent any existing methodology, and (3) develop or tailor methodologies for specific situations. These three goals have guided our initial analysis of the model. Next, we look at criteria for analyzing methodologies and demonstrate how the model helps to assess the assumptions and weaknesses of two existing methodologies. We then provide a detailed example of how one step of the Collaborative Software Engineering Methodology [7] can be represented using the model. Finally, we briefly summarize the results of our interviews with expert analysts and compare the model to the experts’ approaches to tailoring their elicitation methodologies.

**Using the Model to Analyze Existing Methodologies**

A complete and correct elicitation methodology must include a variety of attributes:

- **Completeness.** Each step of a methodology must make assumptions about the state of the requirements and current situation. If these conditions are not documented, the methodology is incompletely specified.
- **Applicability.** Each step of a methodology makes assumptions about the state of the requirements and current situation. Many of these preconditions are the direct outcome of previous steps, and thus are “safe” assumptions. However, in some cases these preconditions are not the result of previous steps, and thus may not be true. If so, the step can be characterized as shown in Figure 6(b), and the methodology is inapplicable; that is, $t_i \notin \sigma(R_i, S_i, \chi(T))$.
- **Necessity.** Although the values of $R_i$ and $S_i$ before a step may represent necessary and sufficient conditions for the next step to transpire, these values are also necessary and sufficient to make valid many other next steps besides the one recommended.3

Therefore, to be useful to the analyst, the definition of each step in a methodology should describe:

1. What requirements must be known (i.e., are you ready to apply this step)? $R_i$ in Figure 3(b).
2. What situations exist (i.e., will this step work right now)? $S_i$ in Figure 3(b).
3. What you should do to perform this step? \( t_i \) in Figure 3(b).
4. What outputs are produced as a result of performing this step (i.e., what requirements will be known after performing the step)? \( R_{t_i} \) in Figure 3(b).
5. What new conditions will be true as a result of performing this step? \( S_{t_i} \) in Figure 3(b).

It is interesting to note that most documented elicitation methodologies start with the assumption that no requirements are known (1), omit any discussion of the situation that exists (2) or that will result (5) altogether, and say little to nothing about how to elicit the information (3) other than to say “produce x” (4). The limited nature of the guidance provided by most methodologies results in a partially defined elicitation process as shown in Figure 7.

For example, one popular methodology goes to great lengths to precisely define the series of requirements models produced by the analyst. Thus, the \( R_s \) are all defined, but the applicable situations (e.g., potential or actual users must be identifiable) are ignored as well as a description of what the analyst should do in order to produce those models. The same is true of the original Structured Analysis [8], which is still used today [20], and which carefully defines the format and content of four critical requirements models (current physical, current logical, new logical, and new physical), but fails to address the conditions under which this is applicable (e.g., there must be an existing system), and fails to tell analysts what they should do to create these models. Specifically, both of these methodologies provide a view of elicitation as shown in Figure 7.

Comparing the Model to the Collaborative Software Engineering Methodology

The Collaborative Software Engineering Methodology (CSEM) [7] was created to support incremental development of complex systems with large, diverse user populations. CSEM divides the requirements process into six primary activities (which may be repeated, eliminated, or otherwise tailored based on the project situation) and provides specific recommendations on what techniques and tools can be used to conduct those activities. The purpose of showing this example is to demonstrate how any documented methodology can be described in terms of our model.

The following example describes the first of the six steps of the CSEM requirements process using the terms of our new unified model (additional steps are analyzed in Hickey and Davis [18]). This example makes a variety of assumptions, including:

- That the selector function, \( \sigma \), has already been created. In reality, we have not yet constructed this function (see later section on future research).
- That we are applying CSEM to the first increment of a new, complex information system.
- That the requirements we need to elicit must identify the basic objects, functions, and states [5] of the new system.
The example describes the inputs \((R_i, S_i)\), techniques recommended by our model, and outputs \((R_{i+1}, S_{i+1})\) generated by use of the selected technique. We then compare our model’s recommendations to those of the CSEM.

Elicitation Step 1

1. Assess requirements \((R)\). Since the project has just begun, we only have a broad definition of project scope that provides a high-level description of the desired functions. We do not yet have a common understanding of the business functions, nor have we identified which specific business functions will be included in the project’s scope.

2. Assess situation \((S)\). We have a large, diverse user population with common goals, but unique operating environments and a variety of legacy systems providing some of the desired functionality. We have identified a representative group of users who can travel to a face-to-face meeting.

3. Select technique \((t)\). The technique selection process \(\sigma\) identifies several possible techniques based on the above characteristics:
   - collaborative workshops rate high because of the diversity of users and the ability to gather representatives in one place;
   - various activity decomposition and modeling techniques (e.g., IDEF0) also rate high because of the need to have a common view of business activities/functions;
   - interviews are another possibility if time is not an issue and users generally agree on functions.

We apply our personal preferences and select a collaborative workshop to develop a simple activity/function hierarchy and agree on which functions are included in the scope of the project.

4. Perform \(\text{elicit}(R, S, t) \rightarrow R_{i+1}, S_{i+1}\). In this case, let us assume that the collaborative workshop succeeded in eliciting the new system’s functions. User representatives now share a common understanding of those functions and agree on which functions will be included in the first increment of the new system.

5. CSEM comparison. The above recommendations are comparable to the first CSEM requirements activity, identify business activities, which recommends use of a group support system (GSS) tool to collaboratively develop an activity hierarchy or more complete IDEF0 activity model. This shows that CSEM’s first activity \(t_1\) is in fact an element of \(T_1\).

\[ R_1 \rightarrow R_2 \rightarrow R_3 \rightarrow R_4 \rightarrow R_{n-1} \rightarrow R_n \]

Figure 7. A Partially Defined Elicitation Process.
The preceding example demonstrates a variety of items:

1. We can use it to analyze the CSEM methodology. For example, the model makes it explicit what (perhaps) tacit assumptions CSEM is making at each step. Thus, our model could be used by CSEM researchers to better understand their own methodology.

2. We can use it to adapt the CSEM methodology. Thus, a user of CSEM could utilize our model to determine (a) the applicability of CSEM to their situation, (b) reasonable alternatives to the defined steps of CSEM, and (c) optimal selections among the alternative techniques recommended as part of the tailoring advice of CSEM.

3. We can use it to analyze our model. As we study CSEM and many other methodologies using our unified model of elicitation, we will likely discover weaknesses in our model.

4. We can use it to compare and contrast the assumptions made by CSEM versus those of other methodologies.

Comparing the Model to Expert Analysts’ Processes

To better understand how expert analysts select elicitation techniques, we are in the process of conducting in-depth interviews with expert analysts who have extensive requirements elicitation experience. To date, we have interviewed Grady Booch, Larry Constantine, Tom DeMarco, Don Gause, Ellen Gottesdiener, Tim Lister, Lucy Lockwood, Don Reifer, Suzanne Robertson, Karl Wiegers, and Ed Yourdon. Combined, these 11 experts have over 285 years experience analyzing requirements on more than 700 projects, working with companies around the world. Detailed results from nine of these interviews are discussed in Hickey and Davis [20]. The purpose of this discussion is to compare the experts’ approaches to requirements elicitation and elicitation technique selection to the approach described by our unified model. Overall, the two approaches were extremely compatible. Specifically,

- All the experts specifically validated the iterative nature of elicitation. This supports modeling elicitation as a series of steps, \(elicit_1, elicit_2, \ldots, elicit_n\).
- When telling stories about specific elicitation projects, many of the experts would say things like “since I know this” or “I need to know this,” and would vary their elicitation methodology based on the type of information that were seeking. This supports the inclusion of the current state of the requirements \(R_i\) in the model.
- All the experts varied the elicitation techniques they used based on the situation. They specifically mentioned a wide variety of problem domain, solution domain, and project characteristics that influenced their selection of techniques. This supports the inclusion of the situation \(S_j\) in the model.
- When describing why they chose a technique, the experts often described specific characteristics of the technique that made it suitable for that situation. This supports the inclusion of the characteristics of techniques \(\chi(T)\) in the model.
The previous three observations also support the inclusion of the selector function $\sigma$ in the model, as well as its functional dependence on $R$, $S$, and $\chi(T)$.

Each of the experts seemed to have preferred, or default techniques that they routinely used. However, as described above, they did vary their approach when the situation changed. This indicated that while there clearly is a preference function $\pi$, experts were familiar with enough techniques that they were able to select alternative techniques that met the changing situational requirements as needed.

Future Research

The model defined in this paper has become the basis for a myriad of new research directions. A few are introduced here:

- **Taxonomy of problem, solution, and project characteristics.** This paper highlights the critical role played by situational characteristics, $S$. We have developed an initial taxonomy of situational characteristics [19] based on earlier taxonomies documented in Alexander and Davis [1], Jørgensen [26], Juristo and Moreno [27], and Scharer [39], but used for quite different purposes. We plan to use the results of interviews with expert analysts to verify or expand this taxonomy.

- **Taxonomy of requirements elicitation techniques.** We have also developed an initial taxonomy of elicitation techniques [19]. We need to compare this taxonomy to the problem, solution, and project situational characteristics to ensure the two taxonomies are compatible. That is, if two elicitation techniques are applicable in the same situation, they should appear in the same place in the taxonomy. We also plan to verify the taxonomy against the results of the expert interviews.

- **Implementation of the selector function.** We have begun the implementation of a system that accepts as input current situational characteristics (i.e., those that capture the problem domain, solution domain, and project domain), the state of the known requirements, and outputs the set of applicable elicitation techniques. We will use the results of the expert interviews to refine and validate the implementation of the selector function. As its use becomes more widespread, the resultant sharing of best practices will assist in the increased success record for software development projects worldwide.

Summary

This paper has introduced a new unified model of requirements elicitation and elicitation technique selection that highlights the critical knowledge required by, and defines the underlying bases of, these disciplines. This model enables:

- all (not just the most experienced) analysts to better understand the requirements elicitation process, the knowledge required to perform elicitation, and the factors they should consider when selecting elicitation techniques;
• analysts to compare and contrast the assumptions and results achieved by the use of any elicitation methodology;
• analysts to easily tailor existing methodologies for unique situations;
• analysts to no longer be bound by predefined methodologies, but, instead, to create new elicitation methodologies easily, by defining situational characteristics and then observing and recording the resultant instances of methodologies;
• project managers to have an improved appreciation for the complexity of requirements elicitation and the importance of selecting appropriate elicitation techniques to ensure effective elicitation and enhance the probability of project success; and
• researchers to explicitly state the assumptions their elicitation methodologies are making about the situation at every step.

Acknowledgments: This is a substantially revised version of a paper and workshop presented at the Thirty-Sixth Hawaii International Conference on System Sciences. The authors thank the conference participants and reviewers for their many helpful suggestions for extending and improving this paper. They are also grateful to the Colorado Institute for Technology Transfer and Implementation (CITTI) at the University of Colorado at Colorado Springs for its financial support of a portion of this research.

NOTES
2. This model represents a conceptual view of the parallel nature of requirements activities. The relative size, shape, and sequence of requirements activities may vary significantly from project to project based on the systems development methodology used as well as a wide variety of project characteristics and other situational factors.
3. The actual mapping from situational characteristics to recommended elicitation techniques is beyond the scope of the current paper. Interested readers are encouraged to learn these actual mappings by reading Hickey and Davis [20].

REFERENCES


