

## 2 Requirements Elicitation: A Survey of Techniques, Approaches, and Tools

Didar Zowghi and Chad Coulin

**Abstract:** Requirements elicitation is the process of seeking, uncovering, acquiring, and elaborating requirements for computer based systems. It is generally understood that requirements are elicited rather than just captured or collected. This implies there are discovery, emergence, and development elements to the elicitation process. Requirements elicitation is a complex process involving many activities with a variety of available techniques, approaches, and tools for performing them. The relative strengths and weaknesses of these determine when each is appropriate depending on the context and situation. The objectives of this chapter are to present a comprehensive survey of important aspects of the techniques, approaches, and tools for requirements elicitation, and examine the current issues, trends, and challenges faced by researchers and practitioners in this field.

**Keywords:** requirements, elicitation, techniques, approaches, tools, issues, challenges, trends, survey.

### 2.1 Introduction

The importance of requirements engineering (RE) within software systems development has long been established and recognized by researchers and practitioners alike (Chapter 1). The elicitation of requirements represents an early but continuous and critical stage in the development of software systems. The requirements for a software system may be spread across many sources. These include the problem owners, the stakeholders, documentation, and other existing systems. Because of the communication rich nature of requirements elicitation activities, many of the effective techniques do not originate from the traditional areas of software engineering or computer science research. Techniques for requirements elicitation are derived mostly from the social sciences, organizational theory, group dynamics, knowledge engineering, and very often from practical experience.

The process of requirements elicitation is generally accepted as one of the critical activities in the RE process. Getting the right requirements is considered as a vital but difficult part of software development projects [36]. A recent field study of fifteen RE teams carried out by Hofmann and Lehner [31] identified key RE practices that should lead to project success. Effective elicitation of requirements was arguably among the most important of the resulting recommended good RE practices.

Requirements elicitation itself is a very complex process involving many activities, with multiple techniques available to perform these activities. The multi-

disciplinary nature of requirements elicitation only adds to this complexity. Elicitation is subject to a large degree of error, influenced by key factors ingrained in communication problems. Despite the importance of requirements elicitation within software development, insufficient attention has been paid to this area in industry and software engineering research to date.

In reality requirements elicitation is a multifaceted and iterative activity that relies heavily on the communication skills of requirements engineers and the commitment and cooperation of the system stakeholders. One of the main problems facing software development project teams is communication barriers and agreement about the requirements. The main point is that concepts that are clearly defined to one community of participants can be entirely opaque to members of another. The fact that this situation exists often goes unnoticed in the course of elicitation unless specific attention is paid to the problem.

The type of the system and the purpose of the project significantly affect the way in which requirements elicitation is conducted. For example it can be said that the method employed for a custom built embedded control system is likely to be substantially different to that of a commercially available inventory management system. The elicitation of requirements can be performed in a variety of settings including the development of web based information systems (Chapter 15) and market driven product lines (Chapter 13), the implementation of large enterprise systems, the selection of commercial off the shelf products (COTS), and the maintenance of existing and legacy systems. Furthermore project teams may be spread across different geographical locations and from diverse cultural backgrounds. The specific elicitation techniques used for a particular situation often depend on a variety of additional factors including time and cost, the availability of resources, the safety criticality of the system, and any legal or regulatory constraints.

In this chapter we present the state of the art and practice in requirements elicitation through an extensive review and analysis of the relevant literature bearing in mind the interdisciplinary and practical nature of this important activity. The aim is to inform the reader of the strengths and weaknesses of some of the current techniques, approaches, and tools used in requirements elicitation today.

The chapter is structured as follows: Section 2.2 introduces the process of requirements elicitation, the activities associated with it, and the roles performed during elicitation by the analyst. Section 2.3 surveys a wide variety of techniques and approaches used for requirements elicitation, and includes a comparison of these with respect to each other and the activities they are used for. Section 2.4 provides some examples of methodology based requirements elicitation, and Section 2.5 presents the types of available tool support for this process. Section 2.6 describes some of the most common issues and pitfalls experienced during requirements elicitation, and Section 2.7 is dedicated to the current trends and challenges in this field. Section 2.8 offers some suggestions for future directions in requirements elicitation research, and finally Section 2.9 contains a brief summary of the chapter.

## 2.2 What is Requirements Elicitation?

Currently there is very little uniformity in RE research and practice concerning a standard definition for requirements elicitation. Requirements elicitation is all about learning and understanding the needs of users and project sponsors with the ultimate aim of communicating these needs to the system developers. A substantial part of elicitation is dedicated to uncovering, extracting, and surfacing the wants of the potential stakeholders. Robertson and Robertson [54] refer to this process as “trawling for requirements” to highlight the fact that through this process you are likely to get more requirements than expected. This implies that gathering a few extraneous requirements initially is always better than gathering less. This is one of the reasons why prioritization (Chapter 4) and negotiation (Chapter 7) are important parts of RE, especially within market driven RE (Chapter 13) where an overload from the constant influx of large amounts of requirements is a serious issue (Chapter 10). More recently the concepts of *inventing* and *creating* requirements have been used to highlight the role of creativity and to emphasize what really goes on during requirements elicitation [43].

### 2.2.1 The Process of Requirements Elicitation

The requirements elicitation process involves a set of activities that must allow for communication, prioritization, negotiation, and collaboration with all the relevant stakeholders. It must also provide strong foundations for the emergence, discovery, and invention of requirements as part of a highly interactive elicitation process. Requirements elicitation involves activities that are intensely communicative. These activities increase in significance when one considers the “culture gap” [62] or basic semantic differences dividing the problem owning and the problem solving communities when attempting to engage in meaningful dialogue [7].

Once again there is very little uniformity in the research literature and practice concerning the names given to the activities often performed during requirements elicitation. However what is generally accepted is that elicitation is the initial stage within the RE process albeit an iterative and integrated one. Typical activities of the requirements elicitation process can be divided into five fundamental types as described below:

- **Understanding the application domain** – It is important when beginning the process of requirements elicitation to investigate and examine in detail the situation or “real world” in which the system will ultimately reside (sometimes called the application domain) [34, 68]. The current environment needs to be thoroughly explored including the political, organizational, and social aspects related to the system, in addition to any constraints they may enforce upon the system and its development. Existing work processes and the related problems to be solved by the system need to be described with respect to the key business goals and issues.

- **Identifying the sources of requirements** – Requirements may be spread across many sources and exist in a variety of formats [41]. In all software development projects a number of possible sources for requirements may be identified. Stakeholders represent the most obvious source of requirements for the system. Users and subject matter experts are used to supply detailed information about the problems and user needs. Existing systems and processes represent another source for eliciting requirements, particularly when the project involves replacing a current or legacy system. Existing documentation about the current systems and business processes including manuals, forms, and reports can provide useful information about the organization and environment, as well as requirements for the new system and their supporting rationale and importance.
- **Analyzing the stakeholders** – Stakeholders are people who have an interest in the system or are affected in some way by the development and implementation of the system and hence must be consulted during requirements elicitation. Typically stakeholders include groups and individuals internal and external to the organization. The customer, and more specifically the project sponsor, is usually the most apparent stakeholder of the system. In some cases however the actual users of the system may be the most important. Other parties whose sphere of interest may extend to some part of the system operations, such as those responsible for work process standards, customers, and partners, should also be regarded as stakeholders if affected. One of the first steps in requirements elicitation therefore is to analyze and involve all the relevant stakeholders. An extensive list of potential project stakeholders that should be consulted during this activity is available in the literature (e.g. [3, 54]). The process of analyzing the stakeholders also often includes the identification of key user representatives and product champions.
- **Selecting the techniques, approaches, and tools to use** – Although some may advocate that just one elicitation technique or a single methodology is sufficient and may be applied to all cases, it is generally accepted that an individual requirements elicitation technique or approach cannot possibly be suitable for all projects. The choice of techniques to be employed is dependent on the specific context of the project and is often a critical factor in the success of the elicitation process [48]. Hickey and Davis [28, 29] have investigated the elicitation technique selection and state that a particular elicitation technique may be selected for a variety of reasons. These include (a) the technique selected is the only one the analyst knows, (b) the technique selected is the analyst's favorite, (c) the selected technique is the one prescribed by a specific methodology that is being followed for the system development, and (d) the choice of technique is governed solely by the intuition of the analyst to be effective in the current context. Clearly requirements elicitation is best performed using a variety of techniques. In the majority of projects several methods are employed during and at different stages in the software development life cycle, often in cooperation where complementary.
- **Eliciting the requirements from stakeholders and other sources** – Once the sources of requirements and the specific stakeholders have been identi-

fied, then begins the actual elicitation of the core requirements using the selected elicitation techniques, approaches, and tools. During this activity it is important to establish the level of scope for the system and investigate in detail the needs and wants of the stakeholders, especially the users. It is also essential to determine the future processes the system will perform with respect to the business operations, and examine the ways in which the system may support them in order to satisfy the major objectives and address the key problems of the business.

It is important to remember that requirements elicitation does not occur in a vacuum. It is strongly related to the context in which it is conducted and specific characteristics of the project, organization, and environment [11]. In practice the budget and schedule of the project have a significant effect on the process and the way in which it is performed. The structure and maturity of the organization will determine how requirements are elicited, as will the way in which the system will interact with users and other systems. The level of volatility within a project must also be considered, as this will directly affect the quality of requirements and the elicitation process itself.

Typically the process begins with an informal and incomplete high-level mission statement for the project [69]. This may be represented by a set of fundamental goals, functions, and constraints for the target system, or as an explanation of the problems to be solved. In order to develop this description, stakeholders and other sources of requirements are identified and used for elicitation. These preliminary results form the basis of further investigation and refinement of requirements in a typically iterative and incremental manner.

Over the years a number of process models have been proposed for requirements elicitation [13, 39, 58]. For the most part these models provide only a generic roadmap of the process with sufficient flexibility to accommodate the basic contextual differences of individual projects. The inability of these models to provide definitive guidelines is a result of the wide range of task that may be performed during requirements elicitation, and the sequence of those activities being dependent on specific project circumstances. The variety of issues that may be faced and the number of techniques available to use only makes it more complex.

In most cases the process of requirements elicitation is performed incrementally over multiple sessions, iteratively to increasing levels of detail, and at least partially in parallel with other system development activities. In reality its completion is often determined by time and cost constraints rather than achieving the required level of requirements quality and completeness. Typically the result of this process is a detailed set of requirements in natural language text and simple diagrammatic representations with additional information including descriptions of the sources, priorities, and rationales.

### 2.2.2 Roles of the Requirements Engineer during Elicitation

During requirements elicitation the requirements engineer (also sometimes referred to as the systems analyst or business analyst) may play a variety of roles and assume different responsibilities. These responsibilities and roles are dependent on the project, people, context and organization involved. A substantial part of elicitation involves *exploring* the problem domain and the requirements that are situated in that domain. Furthermore the requirements engineers often need to perform some typical aspects of project management. Not only do they have to manage the process of elicitation, but they also have to communicate it effectively to the stakeholders. This involves among other things, decision-making (Chapter 12), prioritization (Chapter 4), and negotiation (Chapter 7).

Requirements engineers often play the important role of *facilitator*. When eliciting requirements by group work sessions, they are not only required to ask questions and record the answers, but must guide and assist the participants in addressing the relevant issues in order to obtain correct and complete requirements information. They are also responsible for ensuring that participants feel comfortable and confident with the process, and are given sufficient opportunity to contribute. This role represents a significant part of the skill and expertise required by the analyst in order to perform effective requirements elicitation.

During elicitation conflicts between elicited requirements and stakeholders themselves are inevitable. In many cases the prioritization of requirements from different stakeholders groups is a source of much debate and dispute. When these situations occur the analyst is often playing the role of a *mediator* and is responsible for finding a suitable resolution through negotiation and compromise. It is important that the analyst is sensitive to all the political and organizational aspects of the project when mediating discussions related to the system.

Frequently requirements engineers are responsible for *documenting* the requirements elicited. This role is particularly important as it represents the production of results from the elicitation process, and forms the foundation for the subsequent project phases. Evaluation of the elicitation process and the work performed by the analyst is based on these resultant artifacts, which in some cases may form the basis of contractual agreements.

Analysts are often required to assume the various roles of the *developer* community during requirements elicitation. This includes system architects, designers, programmers, testers, quality assurance personnel, implementation consultants, and system maintenance administrators. This is often due to the fact that these stakeholders have not yet been assigned to the project at the requirements elicitation stage. Despite this the decisions made during this phase of the project will significantly affect these stakeholders and the subsequent phases of development.

All the requirements elicited must be *validated* against the other stakeholders, other systems, each other, and then compared with previously established goals for the system. By this it is meant that the requirements describe the desired features of the system appropriately, and that those requirements will provide the necessary functions in order to fulfill the specified objectives of the target system.

This process typically involves all the identified stakeholder groups, and results in further elicitation activities.

## **2.3 Techniques and Approaches for Requirements Elicitation**

For over two decades now much of the research and practice within RE for software systems has been largely directed towards improving the complex process known as elicitation through the application and development of various techniques, approaches, and tools. Many of these methods have been borrowed and adapted from other disciplines such as the social sciences, and only a select few have been developed specifically for eliciting software requirements [14].

It is important to explain what we mean by the terms “technique” and “approach” as for each of them there exists a number of different uses for them in practice and multiple definitions in the literature. A “technique” is a way of doing something or a practical method applied to some particular task. An “approach” on the other hand is a systematic arrangement, usually in steps, of ideas or actions intended to deal with a problem or situation.

In reality there are literally hundreds of different techniques and approaches from a variety of sources that can and have been employed for requirements elicitation. Below we present only some of those that are more widely used. Although not exhaustive, we believe this selection is representative of the range described in the literature and practiced in industry today.

### **Interviews**

Interviews [1, 32] are probably the most traditional and commonly used technique for requirements elicitation. Because interviews are essentially human based social activities, they are inherently informal and their effectiveness depends greatly on the quality of interaction between the participants. Interviews provide an efficient way to collect large amounts of data quickly. The results of interviews, such as the usefulness of the information gathered, can vary significantly depending on the skill of the interviewer [23]. There are fundamentally three types of interviews being unstructured, structured, and semi-structured, the latter generally representing a combination of the former two.

Unstructured interviews are conversational in nature where the interviewer enforces only limited control over the direction of discussions. Because they do not follow a predetermined agenda or list of questions, there is the risk that some topics may be completely neglected. It is also a common problem with unstructured interviews to focus in too much detail on some areas, and not enough in others [45]. This type of interview is best applied for exploration when there is a limited understanding of the domain, or as a precursor to more focused and detailed structured interviews.

Structured interviews are conducted using a predetermined set of questions to gather specific information. The success of structured interviews depends on knowing what are the right questions to ask, when should they be asked, and who

should answer them. Templates that provide guidance on structured interviews for requirements elicitation such as Volere [54] can be used to support this technique. Although structured interviews tend to limit the investigation of new ideas, they are generally considered to be rigorous and effective.

### **Questionnaires**

Questionnaires [21] are mainly used during the early stages of requirements elicitation and may consist of open and/or closed questions. For them to be effective, the terms, concepts, and boundaries of the domain must be well established and understood by the participants and questionnaire designer. Questions must be focused to avoid gathering large amounts of redundant and irrelevant information. They provide an efficient way to collect information from multiple stakeholders quickly, but are limited in the depth of knowledge they are able to elicit. Questionnaires lack the opportunity to delve further on a topic, or expand on new ideas. In the same way they provide no mechanism for the participants to request clarification or correct misunderstandings. Generally questionnaires are considered more useful as informal checklists to ensure fundamental elements are addressed early on, and to establish the foundation for subsequent elicitation activities.

### **Task Analysis**

Task analysis [9, 53] employs a top-down approach where high-level tasks are decomposed into subtasks and eventually detailed sequences until all actions and events are described. The primary objective of this technique is to construct a hierarchy of the tasks performed by the users and the system, and determine the knowledge used or required to carry them out. Task analysis provides information on the interactions of both the user and the system with respect to the tasks as well as a contextual description of the activities that take place. In most cases considerable effort is required to perform thorough task analysis, and it is important to establish what level of detail is required and when components of the tasks need to be explored further.

### **Domain Analysis**

Examining the existing and related documentation and applications is a very useful way of gathering early requirements as well as understanding and capturing domain knowledge, and identification of reusable concepts and components. These types of investigations are particularly important when the project involves the replacement or enhancement of an existing legacy system. Types of documentation that may be useful for eliciting requirements include design documents and instruction manuals for existing systems, and hardcopy forms and files used in the current business processes. Application studies often also include looking at both upstream and downstream systems, as well as competitive or like solutions. In most cases these studies involve other elicitation techniques such as observing the existing system in use and interviewing the current users.

Domain knowledge in the form of detailed descriptions and examples plays an important part in the process of requirements elicitation. Approaches based on this

type of information are often used in conjunction with, and as the input to other elicitation techniques. For example analysts use previous experience in similar domains as a discussion template for facilitating group work and conducting interviews. Analogies and abstractions of existing problem domains can be used as baselines to acquire specific and detailed information, identify and describe possible solution systems, and assist in creating a common understanding between the analyst and stakeholders. These approaches also provide the opportunity to reuse specifications and validate new requirements against other domain instances [61]. Problem Frames [35] in particular provide a method for detailed problems examination in order to identify patterns that could provide links to potential solutions.

### **Introspection**

The technique of introspection [23] requires the analyst to develop requirements based on what he or she believes the users and other stakeholders want and need from the system. Despite being employed by most analysts to some extent, this technique is mainly used only as a starting point for other requirements elicitation efforts. Introspection is only really effective when the analyst is not only very familiar with the domain and goals of the system, but also expert in the business processes performed by the users. In cases where the analyst is forced to use this technique more, for example when the users have little or no previous experience with software systems in their work environment, a type of facilitation introspection should take place via other elicitation techniques such as interviews and protocol analysis.

### **Repertory Grids**

Repertory grids [38] involve asking stakeholders to develop attributes and assign values to a set of domain entities. As a result the system is modeled in the form of a matrix by categorizing the elements of the system, detailing the instances of those categories, and assigning variables with corresponding values to each one. The aim is to identify and represent the similarities and differences between the different domain entities. These represent a level of abstraction unfamiliar to most users. As a result this technique is typically used when eliciting requirements from domain experts. Although more detailed than card sorting, and to a lesser degree laddering, repertory grids are somewhat limited in their ability to express specific characteristics of complex requirements.

### **Card Sorting**

Card sorting requires the stakeholders to sort a series of cards containing the names of domain entities into groups according to their own understanding. Furthermore the stakeholder is required to explain the rationale for the way in which the cards are sorted. It is important for effective card sorting that all entities are included in the process. This is possible only if the domain is sufficiently understood by both the analyst and the participants. If the domain is not well established then group work can be used to identify these entities. Class Responsibility Collaboration (CRC) cards [5] are a derivative of card sorting that is also used to determine

program classes in software code. In this technique cards are used to assign responsibilities to users and components of the system. Because entities represent such a high level of system abstraction, the information obtained from this technique is limited in its detail.

### **Laddering**

When using laddering [30] stakeholders are asked a series of short prompting questions, known as probes, and required to arrange the resultant answers into an organized structure. A primary assumption when employing laddering is that the knowledge to be elicited can actually be arranged in a hierarchical fashion. For this technique to be effective, the stakeholders must be able to express their understanding of the domain and then arrange it in a logical way. This knowledge, which is often displayed using tree diagrams, is reviewed and modified dynamically as more is added. Like card sorting, laddering is mainly used as a way to clarify requirements and categorize domain entities.

### **Group Work**

Group work such as collaborative meetings is a very common and often default technique for requirements elicitation. Groups are particularly effective because they involve and commit the stakeholders directly and promote cooperation. These types of sessions can be difficult to organize due to the number of different stakeholders that may be involved in the project. Managing these sessions effectively requires both expertise and experience to ensure that individual personalities do not dominate the discussions. Key factors in the success of group work are the makeup of participants and the cohesion within the group. Stakeholders must feel comfortable and confident in speaking openly and honestly, and it is for this reason that group work is less effective in highly political situations.

### **Brainstorming**

Brainstorming [50] is a process where participants from different stakeholder groups engage in informal discussion to rapidly generate as many ideas as possible without focusing on any one in particular. It is important when conducting this type of group work to avoid exploring or critiquing ideas in great detail. It is not usually the intended purpose of brainstorming sessions to resolve major issues or make key decisions. This technique is often used to develop the preliminary mission statement for the project and target system. One of the advantages in using brainstorming is that it promotes freethinking and expression, and allows the discovery of new and innovative solutions to existing problems.

### **Joint Application Development (JAD)**

Joint Application Development (JAD) [65] involves all the available stakeholders investigating through general discussion both the problems to be solved, and the available solutions to those problems. With all parties represented, decisions can be made rapidly and issues resolved quickly. A major difference between JAD and

brainstorming is that typically the main goals of the system have already been established before the stakeholders participate. Also JAD sessions are typically well structured with defined steps, actions, and roles for participants (including a specialist facilitator). The focus of this type of meeting tends to often be on the needs and desires of the business and users rather than technical issues.

### **Requirements Workshops**

Requirements workshop [25] is a generic term given to a number of different types of group meetings where the emphasis is on developing and discovering requirements for a software system. There are many different forms of requirements workshops including cross functional which involves different types of stakeholders from various areas of the business, Co-operative Requirements Capture (CRC) [42] where like JAD there is a defined set of activities and the development community is especially involved, and Creativity [43] which encourages innovative thinking and expression. Another variation of requirements workshops often used in market analysis is the Focus Group [40].

### **Ethnography**

Ethnography [4, 60] being the study of people in their natural setting involves the analyst actively or passively participating in the normal activities of the users over an extended period of time whilst collecting information on the operations being performed. These techniques are especially useful when addressing contextual factors such as usability, and when investigating collaborative work settings where the understanding of interactions between different users with the system is paramount. In practice, ethnography is particularly effective when the need for a new system is a result of existing problems with processes and procedures, and in identifying social patterns and complex relationships between human stakeholders.

### **Observation**

Observation is one of the more widely used ethnographic techniques. As the name suggests the analyst observes the actual execution of existing processes by the users without direct interference. This technique is often used in conjunction with others such as interviews and task analysis. As a general rule ethnographic techniques such as observation are very expensive to perform and require significant skill and effort on the part of the analyst to interpret and understand the actions being performed. The effectiveness of observation and other ethnographic techniques can vary as users have a tendency to adjust the way they perform tasks when knowingly being watched.

### **Protocol Analysis**

Protocol analysis [23, 46] is where participants perform an activity or task whilst talking it through aloud, describing the actions being conducted and the thought process behind them. This technique can provide the analyst with specific information on and rationale for the processes the target system must support [45]. In

most cases however talking through an operation is not the normal way of performing the task, and as a result may not necessarily represent the true process completely or correctly. Likewise minor steps performed frequently and repetitively are often taken for granted by the users, and may not be explained and subsequently recorded as part of the process.

### **Apprenticing**

Apprenticing [54, 6] involves the analyst actually learning and performing the current tasks under the instruction and supervision of an experienced user. In this technique the analyst is taught the operations and business processes by observing, asking questions, and physically doing, rather than being informed of them, as is the case with protocol analysis. Similar to Role Playing but more involved, apprenticeship is very useful where the analyst is inexperienced with the domain, and when the users have difficulty in explaining their actions. The technique of Emergence takes apprenticeship one step further whereby the analyst becomes actively involved in the real life activities of the business.

### **Prototyping**

Providing stakeholders with prototypes of the system to support the investigation of possible solutions is an effective way to gather detailed information and relevant feedback [60]. It is common that prototypes are used in conjunction with other elicitation techniques such as interviews and JAD. Prototypes are typically developed using preliminary requirements or existing examples of similar systems. This technique is particularly useful when developing human-computer interfaces, or where the stakeholders are unfamiliar with the available solutions. There are a number of different methods for prototyping systems such as storyboards, executable, throwaway and evolutionary, with varying levels of effort required. In many cases prototypes are expensive to produce in terms of time and cost. However, an advantage of using prototypes is that they encourage stakeholders, and more specifically the users, to play an active role in developing the requirements. One of the potential hazards when using prototypes for requirements elicitation is that users may become attached to them, and therefore become resistant to alternative solutions from then on. Despite this the technique is extremely helpful when developing new systems for entirely new applications.

### **Goal Based Approaches**

The fundamental premise of goal modeling (Chapter 9) and goal based approaches is that high-level goals that represent objectives for the system are decomposed (e.g. usually using AND and OR relationships) and elaborated (e.g. with “Why” and “How” questioning) into sub goals and then further refined in such a way that individual requirements are elicited. The result of this process is significantly more complicated and complete than the traditional methods of representing system goals using tree structure diagrams. These approaches are able to represent detailed relationships between domain entities, requirements, and the objectives of the system. In general one of the risks when using goal based approaches is that

errors in the high-level goals of the system made early on can have a major and detrimental follow on effect, and that changing goals are difficult to manage.

In recent times significant effort has been devoted to developing these types of approaches for requirements elicitation such as the F<sup>3</sup> project [8], the KAOS meta model [16] and the i\* framework [67]. The use of goals in conjunction with scenarios to elicit requirements has also attracted considerable attention [55, 51, 26]. In practice these approaches have been particularly useful in situations where only the high-level needs for the system are well known, and there exists a general lack of understanding about the specific details of the problems to be solved and their possible solutions.

### **Scenarios**

Scenarios are widely used in requirements elicitation and as the name suggests are narrative and specific descriptions of current and future processes including actions and interactions between the users and the system. Like use cases, scenarios do not typically consider the internal structure of the system, and require an incremental and interactive approach to their development. Naturally it is important when using scenarios to collect all the potential exceptions for each step.

A substantial amount of work from both the research and practice communities has been dedicated to developing structured and rigorous approaches to requirements elicitation using scenarios including CREWS [15], The Inquiry Cycle [51], SBRE [37], and Scenario Plus [56]. Scenarios are additionally very useful for understanding and validating requirements, as well as test case development.

### **Viewpoints**

Viewpoint approaches aim to model the domain from different perspectives in order to develop a complete and consistent description of the target system. For example a system can be described in terms of its operation, implementation and interfaces. In the same way systems can be modeled from the standpoints of different users or from the position of related systems. These types of approaches are particularly effective for projects where the system entities have detailed and complicated relationships with each other. Viewpoints are also useful as a way of supporting the organization and prioritization of requirements. One common criticism of viewpoint approaches is that they do not enable non-functional requirements to be represented easily, and are expensive to use in terms of the effort required.

Some viewpoint approaches [59, 47] provide a flexible multi-perspective model for systems, using different viewpoints to elicit and arrange requirements from a number of sources. Using these approaches analysts and stakeholders are able to organize the process and derive detailed requirements for a complete system from multiple project specific viewpoints.

### 2.3.1 Comparison of Techniques and Approaches

Two important questions that need to be addressed during requirements elicitation are, (1) which techniques and approaches should be used for a given requirements elicitation activity, and (2) which of these techniques and approaches are complementary or can be used as alternatives. Ultimately each situation is unique and the answers to these questions are highly dependant on the context of the project and system. We acknowledge that because of this there is always the possibility for exceptions to any rule made along these lines, however the following two tables in this section are presented as a way of offering some high level support to this end. The intention is to provide an overview of how different techniques and approaches can be used for each of the requirements elicitation activities, and which of the commonly used techniques and approaches often employed for requirements elicitation can be used in cooperation with, or instead of each other.

Rather than including all the techniques and approaches previously presented in Section 2.3 of this chapter, we have selected a core group of eight techniques and approaches which we believe provide suitable coverage across the spectrum of available techniques and approaches (for example ethnography includes observation, and JAD is an example of groupwork), and that are also appropriately representative of those that are currently both state of the art and state of practice. The information contained in these tables is based largely on our assessment of the literature as well as practical experience and observation in requirements elicitation research and practice.

#### Techniques and Approaches for Elicitation Activities

We have seen that different techniques and approaches have different and relative strengths and weaknesses, and may be more or less suited to particular types of situations and environments. Likewise some techniques and approaches are more appropriate for specific elicitation activities and the types of information that needs to be acquired during those activities. Table 2.1 below presents which of the selected core group of techniques and approaches are best suited (marked with an “X”) for the specific requirements elicitation activities described earlier on in Section 2.2 of the chapter.

**Table 2.1.** Techniques and Approaches for Elicitation Activities.

	Interviews	Domain	Groupwork	Ethnography	Prototyping	Goals	Scenarios	Viewpoints
Understanding the domain	X	X	X	X		X	X	X
Identifying sources of	X	X	X			X	X	X

requirements								
Analyzing the Stakeholders	X	X	X	X	X	X	X	X
Selecting techniques and approaches	X	X	X					
Eliciting the Requirements	X	X	X	X	X	X	X	X

We can see from Table 2.1 above that for each of the requirements elicitation activities there are a number of suitable techniques and approaches that can be used. Apart from interviews, domain analysis, and groupwork, which are generic and flexible enough to provide support for all the listed elicitation activities, goal, scenario, and viewpoint based approaches can also be used extensively throughout the process. Given that we have already classified them as requirements elicitation techniques and approaches, it is natural that all the core techniques and approaches presented in the table can be used for activity of actually eliciting the requirements.

### Complementary and Alternative Techniques and Approaches

In most projects more than one requirements elicitation technique and approach will need to be used, therefore it is useful to select those techniques and approaches that are complementary to achieve the best possible results from the requirements elicitation process. In the same way alternative requirements elicitation techniques and approaches enables greater flexibility to the process, and more choice for the analysts and stakeholders. Table 2.2 below provides some guidance with respect to which of the selected core group of techniques and approaches can be used in cooperation (marked with a “C”), and which can be used as alternatives (marked with an “A”).

**Table 2.2.** Complementary and Alternative Techniques and Approaches.

	Interviews	Domain	Groupwork	Ethnography	Prototyping	Goals	Scenarios	Viewpoints
Interviews		C	A	A	A	C	C	C
Domain	C		C	A	A	A	A	A
Groupwork	A	C		A	C	C	C	C
Ethnography	A	A	A		C	C	A	A

Prototyping	A	A	C	C		C	C	C
Goals	C	A	C	C	C		C	C
Scenarios	C	A	C	A	C	C		A
Viewpoints	C	A	C	A	C	C	A	

We can see from Table 2.1 above that for each of the core requirements elicitation techniques and approaches there are both alternatives and those that are complementary. In some cases, such as when prototypes are operated by users under the observation of the analyst, the combination of these techniques has the potential to provide much richer and more detailed requirements information on both the business processes and the needs of the users. Alternative techniques and approaches are useful if for some reason a selected techniques or approach is not being as effective as expected, or when the analyst is unfamiliar, uncomfortable, or unable to use a particular technique or approach. For example it may not be possible to observe users perform their normal business operations due to the physically hazardous environment in which they work. In this case the analyst may choose to use scenarios to elicit that type of information instead.

## 2.4 Methodology Based Requirements Elicitation

Methodology and model driven approaches (Chapter 3) provide ways of representing the existing or future processes and systems using analytical techniques with the intention of investigating their characteristics and limits. Goal, scenario, and agent based modeling techniques as detailed later in this chapter are also used for requirements elicitation in addition to the two approaches described below.

Structured Analysis and Design (SAD) [19, 66] has been around since the mid nineteen seventies and has been widely written about, promoted, and used. The approach is largely function oriented. It comprises of a collection of techniques such as Data Flow Diagrams (DFD) which detail the functional decomposition with the emphasis on the data in and out of the system and related components, and Entity Relationship Diagrams (ERD) that facilitate the representation of system entities, their attributes, and their relationships to each other. Other SAD techniques used during requirements elicitation include Data Dictionaries and Event Lists.

Object Oriented (OO) approaches, and specifically the Unified Modeling Language (UML) contain several techniques often used for requirements elicitation with established yet flexible notations and formats such as Use Cases diagrams, Use Case descriptions, and Class Diagrams. Use Cases [12] are essentially abstractions of scenarios that describe the functional behavior of the system, and have become especially accepted in both research and practice despite their shortcomings such as impreciseness. The diagrammatic and tabular representations

make them easy to understand and flexible enough to accommodate some context specific information. These techniques are especially effective in projects where there is a high level of uncertainty or when the analyst is not an expert in that particular domain.

Several attempts have been made to develop methodologies that combine a number of techniques with supporting roadmaps and guidelines as a way of addressing requirements elicitation. One such approach of combining techniques suggests that the process should begin with an ethnographic study to discover fundamental aspects of existing patterns and behavior, followed by structured interviews to gain deeper insight into the needs of the stakeholders and the priorities of requirements [23]. Furthermore it is proposed that the more extensive requirements elicitation techniques are used to examine in greater detail those needs deemed important.

In other examples of methodology based approaches, requirements elicitation is a defined but closely integrated activity within other aspects of the software development process, such as is the case with Soft System Methodology (SSM) [10], which addresses organizational problems and change, and Quality Functional Deployment (QFD) [2], which focuses on achieving customer satisfaction through quality based development. Gause and Weinberg [22] on the other hand have developed a methodology centered on requirements elicitation, and provide useful and practical techniques for the process including concepts such as Starting Points and Context-Free Questions.

Agile Methods (Chapter 14) for the most part enforce very little upfront requirements elicitation but instead advocate incremental and iterative discovery throughout and integrated with the software development lifecycle [44]. In addition to interview and prototypes, Agile Methods supports the use of Customer or User Stories. These provide basic descriptions of the business processes and what the system needs to do to support them. Typically these are written on index cards by the customer and used as starting points for the development process. Additional requirements elicited as a result of the process from the ever-present customer are added to a Product Backlog, which represents a living requirements document consisting of prioritized system features and functions.

## **2.5 Tool Support for Requirements Elicitation**

A wide variety of tools exist that have been developed and used to support requirements elicitation. These range from shallow to deep with respect to the level of detail and formality, and from generic to specific in purpose and operation. Tools can support a specific technique or process, and may have varying levels of task automation and assistance. Much like the techniques and approaches described above, some of the tools detailed below have been developed for purposes other than requirements elicitation but applied to it, whereas other have been designed specifically for it. By “tool” we refer to an implement, such as software or an artifact, used in practice to accomplish some act, in this case being require-

ments elicitation. For the most part the use of tools for requirements elicitation has been relatively limited and the more successful applications have tended to be domain or approach specific, with the exception of process guidelines and prototyping utilities.

Templates such as IEEE Std 830 Software Requirements Specification [33] and Volere Requirements Specification Template [54] represent the most basic type of tool used by analysts to support the process of requirements elicitation. In a similar way requirements management tools like DOORS, CaliberRM and RequisitePro provide format based support for the elicitation of requirements. Many analysts also utilize specific modeling tools to assist the process of requirements elicitation. These typically have an easy to use graphical or tabular notation.

A number of tools have been developed to support specific requirements elicitation approaches however so far the mainstream software engineering community has largely not adopted these. Examples include ObjectiveR for goal based modeling, and ART-SCENE for scenario elicitation. Several tools have been developed with cognitive support for the requirements elicitation analyst in mind such as The Requirements Apprentice [52], ACME/PRIME [20], and AbstFinder [24]. Enhanced multimedia support for this process and distributed stakeholders was also identified and addressed by several tools including AMORE [64].

Groupware represents a very wide range of tools that has been applied to requirements elicitation. This covers everything from basic support tools such as discussion boards and video conferencing to generic meeting tools like mind mapping and idea capture software, all the way through to virtual collaboration environments specifically designed groups sessions such as developed by TeamWave [27] and GroupSystems [63].

## **2.6 Issues and Pitfalls of Requirements Elicitation**

There has been little doubt in the past about the complexity and difficulty of requirements elicitation in most situations, but the question is why is this still the case today? Part of the reason is the number of problems that may need to be addressed and overcome during the process of requirements elicitation. In general terms there are a large number of contextual, human, economic, and educational factors which effect and may inhibit effective requirements elicitation. For the sake of explanation we have categorized some of the more commonly occurring issues and pitfalls in requirements elicitation faced by both practitioners and researchers according to the aspect of requirements elicitation that they most relate to. These have been collected from a variety of sources in the literature [1, 28, 49] as well as from practical experience and observation.

### **Process and Project**

Each project is unique and no two requirements elicitation situations are ever exactly the same. The process can be performed as part of a custom software development project, COTS selection activity, product line definition, and existing sys-

tem maintenance operation. Projects can range all the way from simple bespoke web-based applications to large and complex enterprise information system product lines. The environment in which the process takes place can also vary greatly including the geographic distribution of stakeholders and the familiarity of users with software systems. Furthermore the process of requirements elicitation is inherently imprecise as a result of the multiple variable factors, vast array of options and decision, and its communication and socially rich nature. Arguably the most common project based requirements elicitation issue is that the initial scope of the project has not been sufficiently defined, and as such is open to interpretations and assumptions. Projects like all functions of a business are subject to change and influence from internal or external factors including economic, political, social, legal, financial, psychological, historical and geographical.

### **Communication and Understanding**

It is common that stakeholders have difficulty articulating their requirements. In some cases this may be as a result of the analyst and stakeholders not sharing a common understanding of concepts and terms, or the analyst is unfamiliar with the problem. Often stakeholders will have difficulty seeing new ways of doing things, or do not know the consequences of their requirements and as such may not know what is feasible or realistic. Stakeholders may understand the problem domain very well, but are unfamiliar with the available solutions and the way in which their needs could be met. Alternatively stakeholders sometimes suggest solutions rather than requirements. Things that are trivial or constantly repeated by stakeholders are often assumed and overlooked although they may not be apparent to the analyst and other stakeholders.

### **Quality of Requirements**

The requirements elicited may not be feasible, cost-effective, or easy to validate. In other cases they can be vague, lacking specifics, and not represented in such a way as can be measured or tested. Furthermore requirements may be defined at different and insufficient levels of detail. Because the process of elicitation is informal by nature, a set of requirements may be incorrect, incomplete, inconsistent, and not clear to all stakeholders. The context in which requirements are elicited and the process itself is inherently volatile. As the project develops and stakeholders become more familiar with the problem and solution domains, the goals of the system and the wants of the users are susceptible to change. In this way the process of elicitation can actually cause requirements volatility and therefore affect the quality of the requirements as a whole.

### **Stakeholders**

Conflicts between stakeholders and their requirements are common and almost inevitable. Furthermore stakeholders may not want to compromise or prioritize their requirements when these conflicts occur. Sometimes stakeholders do not actually know what they want or what their real needs are, and are therefore limited in their ability to support the investigation of possible solutions. Likewise stakeholder can

be adverse to the change a new system may introduce and therefore have varying levels of commitment and cooperation towards the project. Often stakeholders do not understand or appreciate the needs of the other stakeholders and might only be concerned with those factors that affect them directly. Like all humans, stakeholders can change their minds independently, or as a result of the elicitation process itself.

### **Analyst**

Analysts may not be equipped with sufficient implementation expertise and experience to prepare for and perform effective requirements elicitation including appropriate technique selection and the identification of all requirements sources. This may be as a result of lack of education in terms of theory behind techniques and approaches, or the practice of using soft skills such as listening, communicating, and questioning. Analysts from traditional software engineering backgrounds may sometimes focus on the solution not the problem, and rely on only those techniques they are familiar with for all situations. It is also the case that many analysts do not employ any structured or rigorous processes within software development projects to address requirements elicitation.

### **Research**

It is arguable that many of the available techniques are not sufficiently useful or practical, and the transfer of knowledge required to introduce these methods and approaches to industry is too difficult. In fact the quantity of detailed process guidelines with appropriate tool support is very limited, especially with respect to technique selection and addressing the contextual factors in different situations. This can largely be attributed to the absence of sufficient empirical research, case studies and experience reports on the specific topic of requirements elicitation in the literature. Furthermore, there are no agreed metrics by which to measure the performance of the requirements elicitation process within a software development project.

### **Practice**

In general terms there is still a lack of sufficient awareness, understanding, and expertise in requirements elicitation practice. Large gaps exist between requirements elicitation theory and practice, as well as novice and expert analysts. The result of which is that many are still making the same mistakes time and time again with respect to requirements elicitation and do not acknowledge the real issues and their subsequent effects. It is unfortunate that in many cases organizations and particularly customers are resistant to investing the appropriate time and effort into the process despite an increased need for project success.

## **2.7 Trends and Challenges in Requirements Elicitation**

Over the years a number of important trends and challenges have emerged within the field of requirements elicitation in research and practice although not necessarily the same for both. For that reason we have divided the following section into four areas, namely (1) trends in research, (2) trends in practice, (3) challenges in research, and (4) challenges in practice. These trends and challenges show how the field has progressed and changed, and what still needs to be done to further evolve this process in research and practice.

### **2.7.1 Trends in Requirements Elicitation Research**

As the field of RE began to develop, researchers and practitioners identified that the elicitation of requirements for software-based systems had some unique and complicated characteristics, and therefore needed to be addressed as a new and separate topic from traditional knowledge acquisition [17, 23]. As a result and for a time attention was directed to the development of specific tools and techniques to support this process in the hope of reducing its complexity and resolving some of the key challenges in its execution [52,20]. In the mid to late nineteen nineties the focus of requirements elicitation research however was strongly on developing structured and rigorous manual approaches based on new and different paradigms as opposed to tools. These included those based on goals [16], scenarios [51], viewpoints [59], and domain knowledge [61], which continue to be used today.

Recently the development of much needed support for this process has once again been focused on creating tools but this time for the implementation of those newly developed manual approaches, in addition to adapting generic applications to requirements elicitation such as template-driven documentation generation and assistive groupware applications. This has evolved as a result of the continuing need for improvement and the enduring complexity of the process. Furthermore new approaches to requirements elicitation are being developed to support current and specific topics in software engineering such as agent and aspect oriented methodologies, web based systems, and product lines. Agile methods continue to gain interest and support and subsequently work has been directed to investigating how the requirements elicitation process can be effectively implemented with these techniques whilst still maintaining the fundamental principles.

### **2.7.2 Trends in Requirements Elicitation Practice**

Unfortunately RE is not universally practiced as a distinct phase in software development, however its adoption has been on the steady increase particularly over the past decade or so. Many software organizations have discovered that it is in their best interests and the interests of their customers to invest the required time and effort into this phase by implementing a sufficient degree of structure and

rigor to the process, however for the most part this is only true for the larger and more technically mature organizations.

Overall the majority of analysts assigned the responsibility of eliciting requirements for software systems still use generic and traditional techniques such as interviews and group meetings, and only attempt to use others that they are familiar and comfortable with regardless of the circumstances. In recent times however approaches that have been developed specifically for requirements elicitation, such as JAD, Use Cases, Goal and Scenario based approaches, have grown in popularity and usage at least among experienced practitioners.

The adoption of Agile Methods and modeling approaches such as UML continues to grow with widespread acceptance of use case diagrams and descriptions. The concept of just enough requirements engineering and subsequently elicitation as proposed by Davis [18] has been readily accepted by industry and will hopefully lead to the adoption of robust requirements elicitation without unnecessarily committing to expensive and overly detailed processes.

### **2.7.3 Challenges in Requirements Elicitation Research**

One of the key challenges for researchers remains the development of ways to reduce the infamous gap [57] between research and practice in terms of awareness, acceptance, and adoption. This can only be achieved by establishing the results in practice and making the approaches more attractive, thereby providing the proof and motivation for practitioners to use them. In order to make this happen, researchers need to reduce the complexity of approaches, and expertise required to integrate them into practice. Packaging them into manageable and flexible components with appropriate tool support can facilitate this process.

It is important to work towards reducing the gap between experts and novices through practical roadmaps, frameworks, and guidelines that can be easily taught to students and novices. Finding more efficient and effective ways to transfer expert knowledge is certainly part of this effort. Furthermore educators need to adequately address the wide range of skills and expertise required to produce effective requirements engineers, and provide authentic learning environments for gaining realistic experiences.

Overall research needs to continue to develop ways of improving the process and quality of requirements elicitation, and quantifying its success. Only through application to practice can the true value of new techniques, approaches, and tools be determined.

### **2.7.4 Challenges in Requirements Elicitation Practice**

Industry like academia must also look for ways to reduce the gap between experts and novices by investing time and effort in education on what is currently available, and developing new procedures and process for the transfer of knowledge from senior analyst to juniors. Knowing when and which techniques, approaches

and tools to use combined with the knowledge of how, will ultimately improve the chances of customer satisfaction and project success.

Practitioners need to be able to allocate sufficient time and resources to requirements elicitation. This can be partly achieved by educating customers of the value of being diligent in the process, and presenting the risks of not doing so. It is also important that stakeholders themselves understand the benefits and are committed to process.

Organizations in practice need to be more open to accepting the research results, and prepared to join forces, pool resources, and share information to collaboratively produce improved methods of working, and better results for customers. Industry should be more prepared to address the social and organizational factors involved in requirements elicitation, and focus on building software systems that achieve both the business goals and satisfy the users' needs by using the appropriate techniques.

## **2.8 Future Directions in Requirements Elicitation Research**

Despite the successes and progress to date, many important topics remain open for investigation with respect to providing appropriate techniques, approaches, and tools for requirements elicitation, including specific assistance for novice analysts, cognitive support through intelligent tools, and methods that involve direct interaction with stakeholders. Below we have listed some of the potential requirements elicitation research areas not completely resolved to date that we believe deserve appropriate attention in the coming years:

- Reducing the gap between the theory and practice, and experts and novices
- Increasing the awareness and education of analysts and stakeholders in industry
- Developing guidelines for technique selection and managing the impact of factors on the process
- Investigating ways of collecting and reusing knowledge about requirements elicitation
- Integration and use of new technologies including web and agent based architectures into the next generation of support tools
- Produce and publish case studies and industrial experience reports on how requirements elicitation contributed to successes and failures of projects
- Exploring how requirements elicitation activities relates to new and developing fields of software engineering such as agent based systems, agile development methodologies, and web systems

More collaboration is still required between research and practice in order to fully evaluate the existing approaches, and develop new ones for emerging problems. Many of the best results in requirements elicitation research achieved so far have come from this type of joint work with industry. Awareness and education

remain two of the biggest issues faced for those working in requirements elicitation. Students need to be given practical experience as well as a sound theoretical foundation. Practitioners need to be equipped with a variety of techniques, approaches, and tools to use where appropriate depending on what is best suited to the situation. Customers need to understand the importance of the process, believe in it, and support the efforts involved in doing it right.

## 2.9 Summary

The process of requirements elicitation, including the selection of which techniques, approach, or tool to use when eliciting requirements, is dependant on a large number of factors including the type of system being developed, the stage of the project, and the application domain to name only a few. Because of the relative strengths and weaknesses of the available methods and the type of information they provide, the reality is that in almost all projects a combination of several different techniques will be necessary to achieve a successful outcome. This is supported by the fact that many of the techniques are intended to be used in conjunction with each other, and have complementary attributes as discussed throughout the chapter. Most of the approaches require a significant level of skill and expertise from the analyst to use effectively. However from the range of existing techniques, variations of interviews, group workshops, observation, goals, and scenarios are still the most widely used and successful in practice. Despite attempts to automate parts of the process and develop frameworks and guidelines, requirements elicitation still remains more of an art than a science.

## References

1. Agarwal, R., Tanniru, M.R. (1990): Knowledge Acquisition Using Structured Interviewing: An Empirical Investigation. *Journal of Management Information Systems*, 7(1), pp. 123-140.
2. Akao, Y. (1995) *Quality Function Deployment: Integrating Customer Requirements into Product Design*, Productivity Press: Cambridge, MA.
3. Alexander I.F., Stevens R. (2002) *Writing Better Requirements*, Addison Wesley: Great Britain.
4. Ball, L.J., Ormerod, T.C. (2000): Putting ethnography to work: the case for a cognitive ethnography of design. *International Journal of Human-Computer Studies*, 53 (1), pp. 147-168.
5. Beck, K., Cunningham, W. (1989): A Laboratory For Teaching Object-Oriented Thinking, *Proceedings of the Conference on Object Oriented Programming Systems Languages and Applications*, pp. 1-6, October 1-6, New Orleans, LA.
6. Beyer, H.R., Holtzblatt K. (1995): Apprenticing with the customer. *Communications of the ACM*, 38 (5), pp. 45-52.
7. Bostrum, R.P. (1989): Successful application of communication techniques to improve the systems development process. *Information and Management*, 16 (5), pp. 279-295.

8. Bubenko, J.A., Jr., Wangler, B. (1993): Objectives driven capture of business rules and of information systems requirements, Proceedings of the International Conference on Systems, Man and Cybernetics, pp. 670 – 677, October 17-20, Le Touquet, France.
9. Carlshamre, P., Karlsson, J. (1996): A usability-oriented approach to requirements engineering, Proceedings of the Second International Conference on Requirements Engineering, pp. 145-152, April 15-18, Colorado Springs, CO.
10. Checkland, P., Scholes, J. (1990) *Soft Systems Methodology in Action*, John Wiley & Sons: New York, NY.
11. Christel, M.G., Kang, K.C. (1992): Issues in Requirements Elicitation, Carnegie Mellon University Technical Report, CMU/SEI-92-TR-012.
12. Cockburn, A. (2001) *Writing Effective Use Cases*, Addison Wesley: Reading, MA.
13. Constantine, L., Lockwood, L.A.D. (1999) *Software for Use: A practical guide to the models and methods of usage-centered design*, Addison Wesley: Reading, MA.
14. Coulin, C., Zowghi, D. (2004) Requirements Elicitation for Complex Systems: Theory and Practice, in *Requirements Engineering for Socio-Technical Systems*, edited by Jose Luis Mate and Andres Silva, Idea Group: USA.
15. CREWS, <http://sunsite.informatik.rwth-aachen.de/CREWS/>, Accessed 15 November 2004.
16. Dardenne, A., van Lamsweerde, A., Fickas, S. (1993): GoalDirected Requirements Acquisition. *Science of Computer Programming*, 20 (1-2), pp. 3-50.
17. Davis, A.M. (1994) *Software Requirements: Analysis and Specification*, Prentice Hall: New Jersey.
18. Davis, A.M. (2004) *Just Enough Requirements Management: Where Marketing and Development Meet*, Dorset House: New York.
19. DeMarco, T., Plauger, P.J. (1979) *Structured Analysis and System Specification*, Prentice Hall: New York, NY.
20. Feblowitz, M., Greenspan, S., Reubenstein, H., Walford, R. (1996): ACME/PRIME: Requirements Acquisition for Process-Driven Systems, Proceedings of the Eighth International Workshop on Software Specification and Design, pp. 36-45, March 22-23, Paderborn, Germany.
21. Foddy, W. (1994) *Constructing Questions for Interviews and Questionnaires*, Cambridge University Press: Cambridge.
22. Gause, D.C., Weinberg, G.M. (1989) *Exploring Requirements: Quality before Design*, Dorset House: New York.
23. Goguen, J.A., Linde, C. (1993): Techniques for Requirements Elicitation, Proceedings of the IEEE International Symposium on Requirements Engineering, pp. 152-164, January 4-6, San Diego, CA.
24. Goldin, L., Berry, D.M. (1994): AbstFinder, A Prototype Natural Language Text Abstraction Finder for Use in Requirements Elicitation. *Automated Software Engineering* 4 (4), pp. 375-412.
25. Gottesdiener, E. (2002) *Requirements by Collaboration*, Addison Wesley: Boston, MA.
26. Haumer, P., Pohl, K., Weidenhaupt, K. (1998): Requirements Elicitation and Validation with Real World Scenes. *IEEE Transactions on Software Engineering*, 24 (12), pp. 1036-1054.
27. Herela, D., Greenberg, S. (1998): Using a Groupware Space for Distributed Requirements Engineering, Proceedings of the Seventh Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 57-62, June 17-19, Stanford, CA.

28. Hickey, A.M., Davis, A.M. (2002): The Role of Requirements Elicitation Techniques in Achieving Software Quality, Proceedings of the Eighth International Workshop of Requirements Engineering: Foundation for Software Quality, September 9-10, Essen, Germany.
29. Hickey, A.M., Davis, A.M. (2003): Elicitation Technique Selection: How Do Experts Do It?, Proceedings of the Eleventh IEEE International Requirements Engineering Conference, pp. 169-178, September 8-12, Monterey Bay, CA.
30. Hinkle, D. (1965): The change of personal constructs from the viewpoint of a theory of implications, Ohio State University, Doctoral Dissertation.
31. Hofmann, H.F., Lehner, F., (2001): Requirements Engineering as a success factor in software projects. *IEEE Software*, 18 (4), pp. 58-66.
32. Holtzblatt, K., Beyer, H.R. (1995): Requirements Gathering: The Human Factor. *Communications of ACM*, 38 (5), pp. 30-32.
33. IEEE (1998) IEEE Std 830-1998 Software Requirements Specification
34. Jackson, M. (1995): The World and the Machine, Proceedings of the Seventeenth IEEE International Conference on Software Engineering, pp. 283-292, April 24-28, Seattle, WA.
35. Jackson, M. (2000) Problem Frames: Analyzing and Structuring Software Development Problems, Addison Wesley: Boston, MA.
36. Jones C. (1996) Applied Software Measurement: Assuring Productivity and Quality, McGraw-Hill: New York.
37. Kaufman, L.D., Thebaut, S., Interrante, M.F. (1989): System Modeling for Scenario-Based Requirements Engineering, SERC Technical Report, SERC-TR-33-F.
38. Kelly, G. (1955) The Psychology of Personal Constructs, Norton: New York.
39. Kotonya, G., Sommerville, I. (1998) Requirements Engineering: Processes and Techniques, John Wiley & Sons: Great Britain.
40. Krueger, R.A. (1994) Focus Groups: A practical guide for applied research, Sage: Thousand Oaks, CA.
41. Loucopoulos, P., Karakostas, V. (1995) Systems Requirements Engineering, McGraw-Hill: London.
42. Macaulay, L.A. (1993): Requirements as a Cooperative Activity, Proceedings of the IEEE Symposium on Requirements Engineering, pp. 174-181, January 4-6, San Diego, CA.
43. Maiden, N., Gizikis, A., Robertson, S., (2004): Provoking Creativity: Imagine What Your Requirements Could be Like. *IEEE Software*, 21 (5), pp. 68-75.
44. Martin, R.C. (2003) Agile Software Development: Principles, Patterns, and Practices, Prentice Hall: Upper Saddle River.
45. McGraw, K.L., Harbison-Briggs, K. (1989) Knowledge Acquisition: Principles and Guidelines, Prentice Hall: New Jersey.
46. Nielsen, J., Clemmensen, T., Yssing, C. (2002): Getting access to what goes on in people's heads?: reflections on the think-aloud technique, Proceedings of the Second Nordic Conference on Human-Computer Interaction, pp. 101-110, October 19-23, Aarhus, Denmark.
47. Nuseibeh, B., Finkelstein, A., Kramer, J. (1996): Method Engineering for Multi-Perspective Software Development. *Information and Software Technology Journal*, 38 (4), pp. 267-274.

48. Nuseibeh, B., Easterbrook, S. (2000): Requirements Engineering: A Roadmap, Proceedings of the Conference on The Future of Software Engineering, pp. 35-46, June 4-11, Limerick, Ireland.
49. OPEN Process Framework, <http://www.donald-firesmith.com/>, Accessed 15 November 2004.
50. Osborn, A.F. (1979) Applied Imagination, Charles Scribner's Sons: New York.
51. Potts, C., Takahashi, K., Anton, A.I. (1994): Inquiry-Based Requirements Analysis. IEEE Software, 11 (2), pp. 21-32.
52. Reubenstein, H., Waters, R. (1991): The Requirements Apprentice: Automated Assistance for Requirements Acquisition. IEEE Transactions on Software Engineering, 17 (3), pp. 226-240.
53. Richardson, J., Ormerod, T.C., Shepherd, A. (1998): The Role of Task Analysis in Capturing Requirements for Interface Design. Interacting with Computers, 9 (4), pp. 367-384.
54. Robertson, S., Robertson, J. (1999) Mastering the Requirements Process, Addison Wesley: Great Britain.
55. Rolland, C., Souveyet, C., Ben Achour, C. (1998): Guiding Goal Modeling Using Scenarios. IEEE Transactions on Software Engineering, 24 (12), pp. 1055-1071.
56. Scenario Plus, <http://www.scenarioplus.org.uk/>, Accessed 15 November 2004.
57. Siddiqi J., Shekaran, C. (1996): Requirements Engineering: The Emerging Wisdom. IEEE Software, 13 (2), pp. 15-19.
58. Sommerville, I., Sawyer, P. (1997) Requirements Engineering: A Good Practice Guide, John Wiley & Sons: Great Britain.
59. Sommerville, I., Sawyer, P., Viller, S. (1998): Viewpoints For Requirements Elicitation: A Practical Approach, Proceedings of the IEEE International Conference on Requirements Engineering, pp. 74-81, April 6-10, Colorado Springs, CO.
60. Sommerville, I. (2001): Software Engineering 6<sup>th</sup> Edition, Addison Wesley: USA.
61. Sutcliffe, A., Maiden, N. (1998): The Domain Theory for Requirements Engineering. IEEE Transactions on Software Engineering, 24 (3), pp. 174-196.
62. Taylor-Cummings, A. (1998): Bridging the user-IS gap: a study of major systems projects. Journal of Information Technology, 13 (1), pp. 29-54.
63. Weatherall, A. (1998): Creative problem solving using GroupSystems, Proceedings of the Thirty-First Hawaii International Conference on System Sciences, 1, pp. 588-595, January 6-9, Hawaii.
64. Wood, D., Christel, M., Stevens, S.M. (1994): A Multimedia Approach to Requirements Capture and Modeling, Proceedings of the First International Conference on Requirements Engineering, pp. 53-56, April 18-22, Colorado Springs, CO.
65. Wood, J., Silver, D. (1995) Joint Application Development, John Wiley & Sons: New York.
66. Yourdon, E. (1989) Modern Structured Analysis, Prentice Hall: Englewood Cliffs, NJ.
67. Yu, E.S.K. (1997): Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering, Proceedings of the Third IEEE International Symposium on Requirements Engineering, pp. 226-235, January 5-8, Washington, D.C.
68. Zave, P., Jackson, M. (1997): Four dark corners of requirements engineering. ACM Transactions on Software Engineering and Methodology, 6 (1), pp. 1-30.
69. Zowghi, D. (1999): A Logic-Based Framework for the Management of Changing Software Requirements, Macquarie University, Doctoral Dissertation.

### **Author Biography**

Didar Zowghi is Associate Professor of Software Engineering and the Director of Requirements Engineering Research Laboratory in the Faculty of Information Technology at University of Technology, Sydney. She holds a Bachelors of Science (Hons) and a Masters of Science in Computer Science, and PhD in Software Engineering. She serves on the program committee of many national and international conferences, in particular the IEEE International Conferences on Requirements Engineering since 1998. She is the regional editor (and the editor of the Viewpoints column) of the International Requirements Engineering Journal and the associate editor of the Journal of Research and Practice in Information Technology (JRPIT). She has published extensively on many aspects of Requirements Engineering.

Chad Coulin is a PhD candidate and member of the Requirements Engineering research group in the Faculty of Information Technology at the University of Technology, Sydney. He holds a Bachelors of Engineering (Hons) in Microelectronics, and is currently working on the development of new and innovative methods to support requirements elicitation for software-based systems. His other research interests include Computer Supported Cooperative Work (CSCW) and the development of interactive and intelligent Computer Assisted Software Engineering (CASE) tools. Previously he has worked as a project and product manager in the USA and Europe developing and implementing large-scale industrial software systems.