

# A Social-Creative-Cognitive (SCC) Model for Requirements Engineering

Linda Dawson

**Abstract** This paper describes a research project and the findings from a set of six cases which examine professional requirements engineering (RE) practice from the perspective of how analysts reach agreement on system requirements for information systems development (ISD). In these studies, it was found that the analysts reached agreement on requirements with clients through communication and negotiation based on both analysts' cognitive skills in problem-solving and creative skills in developing informal models that could be used in the negotiation of agreement and sign-off on requirements specifications. A theoretical model of social-creative-cognitive (SCC) aspects of requirements engineering is proposed encompassing the empirical findings.

## 1 Introduction

The social and collaborative aspects of the requirements engineering process have been well documented [7, 17, 18, 24, 26, 27, 36, 38, 41]. Viewing the requirements engineering process as a social process implies that if the product of the requirements engineering process is the specification document on which the design and implementation of the system is based, then this product has to be agreed upon by both parties. That is, the specification needs to be validated as correct or acceptable from both points of view—the formally modelled consultant's point of view and the client's informally modelled point of view.

This paper reports on part of a larger longitudinal project [9, 11] whose objective is to understand how systems are developed in practice and which roles, if any, experience, cognitive processes, social processes, methods, models, tools and

---

L. Dawson (✉)

Faculty of Informatics, University of Wollongong, Wollongong, NSW, Australia  
e-mail: lindad@uow.edu.au

techniques play in the RE process. The research reported in this paper addresses part of the larger project on the basis of the following research question:

How do professional developers reach agreement on requirements for system development?

Sections below present the findings of a multiple-case study addressing these questions based on qualitative data collection and analysis methods. Section 2 provides background in related work. The research approach adopted is presented in Sect. 3 along with descriptions of the six cases. The findings are presented in Sect. 4, and Sect. 5 presents a theoretical model which encompasses the findings and discusses the implications of the findings for practice and education and training.

## 2 Social Processes in Requirements Engineering

Although there is no single commonly accepted definition of requirements engineering [12, 24, 26, 38], the definition used in this paper is

Requirements engineering is an iterative and collaborative process of elicitation, modelling, and validation of information system requirements which provides an agreed specification which is the basis for the design and implementation of that information system.

This definition places the requirements engineering process early in the system development cycle, although in many projects it may not necessarily be the very first phase. It may be that the requirements engineering process is triggered by preliminary investigations [1] or questionings of users and clients of the way activities are undertaken within their organisation [5]. This definition also contains explicit reference to the collaboration needed between clients and analysts [41] and the need for feedback via iteration in the process of specifying requirements.

There are various frameworks proposed for understanding RE practice [20, 24, 26, 31, 32].

Pohl [32] proposes a framework suggesting three dimensions of the requirements engineering process. These three dimensions can be outlined as follows:

- Specification dimension involves the development of the specification from the “opaque” to the specific.
- Representation dimension which deals with the methods for representing the specification and includes informal, semi-formal and formal languages.
- Agreement dimension describes the “common specification” or agreed specification which is based on the different viewpoints of the parties involved in developing the specification.

This view of requirements engineering is in harmony with the idea of information systems development being more than a purely technical undertaking. It incorporates the concept of an information system described by [24] as a “sociotechnical system,” that is, a system “... that involve[s] computer-based components interacting with people and other technical system components in an organisational setting”.

The agreement dimension of Pohl’s framework is specifically explored in this research paper.

### 3 Research Approach

The objective of the project described in this paper was to understand how systems requirements are developed and agreed on in practice and which social or other aspects play a role in this agreement. The research questions are:

- How do professional developers and clients reach agreement on requirements for system development?
- Is this agreement a purely social agreement between the developer and the client?
- What other aspects contribute to this agreement?

The evolutionary case research approach [9] was used in this study based on recorded semi-structured interviews with individual requirements engineers. These interviews provided empirical data which is interpretive and descriptive rather than normative or quantitative. Interview questions focussed on exploring the three main processes of requirements engineering: elicitation, modelling and validation, e.g. Is elicitation explicitly undertaken and when does it start? When does it end? When does modelling begin? Do you think it is necessary to validate the specification once the models have been produced? Which (how many) models are produced during specification? Who are they produced for? Who uses them? Which models, if any, are shown to the user? Which models are used internally by the development team?

The evolutionary case approach is iterative and cyclic and is particularly suited to interview-based data collection. In each case cycle the researcher seeks to refine the current version of the theoretical model by:

- Looking for *reinforcement* of concepts already contained within the theoretical model or framework
- *Revelation*—identifying new areas for exploration and potential reinforcement
- Learning and *reflection* on data collected so far
- *Re-examining* previous transcripts to find any further reinforcement of an emerging theme

The researcher is active in the data collection. Leading questions are encouraged in order to facilitate reinforcement, and semi-structured, open-ended questions are used to facilitate revelation. Exploration of these revelations is incorporated into revised interview scripts which are used in the next case in the cycle. Reinforced concepts are retained in the evolving theoretical model. The process is ongoing but can be concluded when there has been enough reinforcement for a representative model of the research domain being investigated to stand alone or when theoretical saturation has been reached [14]. So, the outcome of the research method is a theory about the area being investigated which is initially grounded in the literature and then progressively grounded in data gained from investigating the application of system development methods in practice.

**Table 1** Background information for each consultant

Case	Job title	Client	Project
1	Operations manager	Federal govt	Complex technical
2	Principal consultant	State govt	Web based transactions
3	Senior consultant	Telecommunications	Fault management system
4	Director and partner	Software developer	Insurance
5	Technical manager	Software developer	Stockbroking package
6	Principal consultant	Hospital	E-prescribing

## 4 The Case Studies

Although the larger longitudinal project currently comprises eight cases, the six cases reported in this study provided data specific to the agreement dimension of requirements engineering. Participants were recruited through industry. Some participants provided contacts for subsequent participants. There was no attempt to select participants based on specific background characteristics. Most participants used object-oriented approaches to ISD, one used an agile feature-driven approach and all participants were familiar with a range of methodologies. The contextual information for each consultant interviewed is summarised in Table 1.

## 5 Findings

Data analysis was cyclic and based on identifying (revealing) and confirming (reinforcing) themes from the interview transcripts using an illustrated narrative style as described by Miles and Huberman [30] and as used in Fitzgerald [15] and Urquhart [41]. Miles and Huberman [30] describe this as looking for "... key words, themes, and sequences to find the most characteristic accounts". Themes that are revealed and reinforced become part of the evolving theoretical model [9].

Case 1 involved a small confidential project which had to be completed quickly for a government department. The system was highly technical and involved complex calculations and predictions. The consulting organisation in this case used a commercial semi-object-oriented method (James [33]; James [34]) based on a template and the use of cards to describe requirements. The template is a booklet that provides guidelines for the tasks which need to be undertaken during the process of requirements specification. Every requirement that is documented is based on a requirements card describing various "characteristics". The cards are filled out in collaboration with the client/users during the requirements specification process. One characteristic associated with agreement is the "fit criteria" which is a user-defined test which ensures a requirement is a single functional unit which can be tested. Also associated with agreement is the customer satisfaction characteristic based on how *happy* the customer would be if the requirement was included and the

customer dissatisfaction characteristic based on how *unhappy* the customer would be if the requirement was not included. This allows the consultant and client to prioritise any “wish list” the client might have.

We write the cards with the client, then we go away and write the document. The cards are essentially self-documenting, but then we do a second level of checking [to ensure] that we haven’t misinterpreted [anything] by actually stepping through it again. The only testing is the identification of the fit criteria.

So, in Case 1 agreement was reached based on the cards developed with the client. The cards represented an informal model of the system that was easily discussed by both parties.

Case 2 used object-oriented methods to develop in-house specification templates representing identified “common transactions” for multiple clients who could then do their own requirements engineering with the assistance of an IT liaison person. Booklets were sent to the client organisations containing instructions on how to use the template, the generic use case (diagram and script), the generic object model, interaction diagram, etc. The first task for the client was to work through the general use case flow diagram to see how well their transaction matched the common model. This test was called a “goodness of fit” test. Customising the template involved modifying the basic flow diagram (based on the “goodness of fit”) and the object model, modifying the use case script by striking out (not removing) elements, so that someone could look across the page and see what had been changed.

So, in Case 2 agreement was based on the client customising a booklet-based template with assistance from an IT person. Only simplified use case diagrams and dialogues were shown to the users when describing requirements. Models based on formal notation were considered too complex for users to understand: “We tell them [the users] that the model is technical mumbo jumbo ... you know I wouldn’t show them a data model either ... the closest I’ve gotten is working with this type of flow diagram (use case flow diagram)... they can follow that pretty well but they don’t usually have the patience to really work through the interaction diagrams or the model.”

The consultant in Case 3 used object-oriented methods to specify and build a fault management system for a telecommunications organisation. In this case agreement was reached using informal models such as use case scripts and a prototype to develop the requirements in consultation with “subject matter experts” nominated by the client. One of the main members of the expert group (the main business contact) was a network manager with about 25 years experience in transmission management. He knew all there was to know about the client’s management of their transmission network. He was involved wherever possible and he played a user liaison role and a business expert role. “A lot of the requirements model was drawn by talking to these guys and verified as well through the development phase.” Knowledge elicitation was done using interviews with users and the special expert user group. It was highly iterative to the degree where the subject matter expert would be calling in every couple of days. “It would have just been a conventional sort of thing, throw some prototype together ... and that can be done very quickly. Get the guy in, sit down and work through our current prototype and that might have

happened once a week for twenty weeks.” Working on this project was one of the subject matter expert’s main job responsibilities. He was freed up from some of his network management responsibilities to come and work with the system team. There was a separate acceptance test suite developed by the users, but that was not set up until well into the development stage approximately six months before acceptance testing was due to start. “All the way through we used our use case model to test things as we were developing ...”.

The consultant in Case 4 had extensive experience in using object-oriented methods to specify and build actuarial and insurance systems. Agreement was reached using models shown to users based on ad hoc diagrams, rich pictures and screen simulations rather than class models or interaction diagrams, although the diagrams using formal notations were used within the team and in the design phase.

In Case 4, the analyst was explicit about using various ad hoc diagrams, pictures, PowerPoint simulations and use cases to reach agreement with the clients/users. “I mean if you draw a picture and that doesn’t make any sense to them then you draw another one ... [a] requirements specification has to be in terms that they understand.” And “... and in every project I’ve ever worked on ... there’s been a few key pictures. The one I’m working on at the moment is the billing cycle—it’s a wheel and its got the steps in the billing cycle on it and that’s in everybody’s head and everybody talks in those terms and it’s just the key base thing—it’s the conceptual core of the thing ... I’m a great believer in ad hoc diagrams that give the picture that springs from your understanding of the problem and in a lot of OO work the process of development hinges on one or two of these pictures.”

The consultant in Case 5 was a senior project manager for a software development organisation which creates custom-built systems for individual clients including generic packaged software systems for the stockbroking industry. The consultant was experienced in many methods, both object-oriented and non-object-oriented, for specifying and building business systems. The methodology was an in-house methodology based on UML notation but not the complete rational development method. Prototyping in the form of a GUI prototype for the users was used in the project. “We actually do a prototype and then work through the users with that and then gain sign off at that level.”

So, in Case 5, agreement was reached using prototypes, screen simulations and animations with use case models used mainly at the validation phase. In this case the prototype was the most used tool for validation and agreement, and use cases were only used for exceptions or special cases “... as we were looking at the requirements document we had the prototype running and projected up on a big screen and we walked through the prototype in relation to the requirements”.

The consultant in Case 6 was a principal consultant in a software development organisation which develops custom-built systems for large healthcare projects. This project was an implementation of the full ICT infrastructure for a new private hospital and also used the hospital as a test bed for the broader healthcare group, to trial new technologies and to roll out systems to the rest of the group. The consultant was experienced in many methods but was using an agile feature-driven approach (AFDD). In this case agreement was reached using text, rich pictures and modified

models “... process modelling with data flows in it, but done in a way that is in a way that the business will understand it, using symbols that that business understands ... rich text, rich pictures”. This consultant also described another method used in a previous project, “... very much the user centric approach, we used the ‘big floppy book’, which was actually a whole heap of explicit, mock screen shots... [p]ut them together as a book and say this is how... your business process works, and they could actually flick through the screens”.

In all the cases reported here, there were two types of models produced: informal models in the form of pictures, text and diagrams, and/or use cases, and prototypes, i.e. models that can be understood and explained without specific training, which were separate from formal models based on specific modelling notations such as entity-relationship modelling and UML modelling. The two types of models were used for different purposes. The informal models were used in the validation of the specification and achieving understanding and agreement with clients, and the formal models were used internally within the analysis team and passed on to the design phase of the development.

## 6 Discussion and Implications

The basis for the perceived need for both informal and formal models in requirements engineering as found in this study confirms that the requirements engineering process is fundamentally a social process involving two main groups: the users/clients and the professional consultants [7, 16, 24, 32, 41]. It is not claimed that the implications discussed here are new or exhaustive, rather that the findings from this research project strengthen the idea that requirements engineering is a social, creative and cognitive process [8, 17, 19, 38].

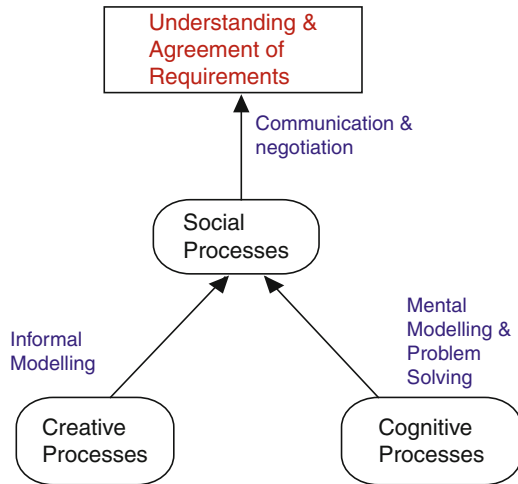
Based on these findings and previous research in the literature as discussed above, we propose a social-creative-cognitive model to encompass the relationships between these concepts. The model is constructed incrementally as follows.

Requirements engineering is a *social process*, and this social process requires understanding by all parties to reach agreement. Understanding requires communication skills, and agreement requires negotiation skills. The facilitation of understanding and agreement also requires *creative modelling skills* on the part on the analyst to produce understandable informal models. These models are developed during elicitation, refined during modelling and used for validation of requirements before sign-off or agreement to go ahead with design, and implementation is given by the client.

Further, creative informal modelling as demonstrated by the analysts in this study relied on *cognitive skills* including abstraction and mental modelling and problem-solving and reasoning skills particularly analogical reasoning skills on the part of the analyst as reported elsewhere in Dawson [10].

These concepts and their relationships are represented in the theoretical model shown in Fig. 1. This model contains features for defining a theoretical model as

**Fig. 1** A theoretical model of the contribution of social, creative and cognitive processes to requirements engineering



defined by Dubin [13] and Bacharach [2], i.e. the interactions or relations between defined units or concepts within a set of boundaries or constraints depicting a limited portion of the world.

In this model, the main social goal of successful requirements engineering is to achieve agreement and understanding about requirements between users/clients and the professional developer or development team. The achievement of this goal depends on three processes. The social process involves the users and the analysts in communication and negotiation which brings about the understanding and agreement. This social interaction is influenced by the professional input of the analyst in the role of problem-solver and modeller. Further, the analyst also has to express the solutions to the problems and the models arrived at in his/her mind in a concrete manner which facilitates the understanding and agreement. This creative process involves the development of informal models (such as diagrams, simulations, animations or textual explanations) that can be understood and discussed by the users and analysts in their social communications and negotiations. These implications are discussed further in the following two sections.

### **6.1 Requirements Engineering as a Social Process**

The perception of the analysts in this study was that for agreement between analyst and client to take place, there needs to be two types of models: informal models for communicating the specification to the user for information and validation and formal models developed by the analyst team to pass on to the design and implementation team.

It has been generally recognised [6, 25, 42] that many of the errors that lead to costly maintenance and/or failure of information systems can be traced to



omissions, inconsistencies and ambiguities in the initial requirements specification. If, as the findings of this research project confirm, the models used for validation of the specification with the clients are different to the models used in design and implementation, then this may indicate one of the areas where these inconsistencies, omissions and ambiguities might arise. Recognising and understanding this issue requires further research and provides a step towards building the right tools and techniques to assist the requirements engineering process.

## ***6.2 Requirements Engineering as a Creative and Cognitive Process***

As with many professional analysis and design activities involving creative reflection [22, 28, 35], requirements engineering can be considered to be a creative process particularly on the part of the analyst producing the requirements specification. The case studies showed evidence of recognition on the part of the practising professionals that they had to be able to model or represent what the users wanted in some diagrammatic form. The findings suggest all of the analysts who used formal notations such as UML or entity-relationship diagrams would not use diagrams based on these notations with the users or clients because they believed the users would not understand them. For all of the analysts reported in this study, this meant that they had to find (considerably diverse) creative solutions to the representation problem: simplified use cases, ad hoc diagrams, rich pictures, animations, PowerPoint simulations and text-based explanations. Each analyst had his/her own creative approach to informal user modelling. There is enough evidence provided in the case studies to imply that this creative approach to user modelling is common in professional requirements engineering practice. This suggests that the variety and use of such informal models should be systematically described in detail, which would be useful to professionals, educators and students.

Closely related to the creative aspects of requirements engineering are the cognitive aspects of requirements engineering. Requirements specification can be considered as a high-level cognitive process [10, 29, 37, 39]. As previously reported [10], in four of the six cases in this study, requirements specification involved mental modelling during the transformation from elicitation to concrete models for design and implementation. Overall, these four analysts believed that they were continually “modelling in the mind” during the elicitation process and that these mental models were further refined in the mind before they were communicated to others (users or fellow analysis team members) or before they were committed to paper. So, there is evidence that requirements modelling requires cognitive skills including abstraction and mental modelling together with problem-solving and reasoning skills particularly analogical reasoning skills on the part of the analyst. These cognitive skills are an essential foundation for the social interactions required between analyst and client for understanding and agreement to be reached.

### 6.3 *Implications for Education and Training*

Typical undergraduate courses in information systems or related disciplines involve some exposure of students to systems analysis methodologies, techniques and tools. Often, students are required to participate in a project where the principles of systems analysis can be applied to an example of an industrial or organisational style system development project. The challenge for academics designing these courses or writing textbooks to accompany these courses is often how to relate theory and project work to real professional practice. Successful programmes in these areas have provided projects with real clients, project team environments and/or other real world environments [3, 4, 21, 23, 40].

The findings from the case studies in this research project have several implications for education and training. Based on the findings presented here, courses seeking to provide realistic commercial and organisational project environments should include the following elements and ideas.

There are many tools and techniques for requirements analysis and specification, and as this research project has shown:

- In practice analysts often develop their own in-house methodologies based on diverse tools and techniques rather than adhere to a single prescribed or commercial methodology.
- Many professional analysts build their own personal methodology by trying out and adapting those techniques and tools that suit their way of thinking and their way of interacting with clients and the particular projects that they are working on.

There are many models for representing requirements, and as this research project has shown:

- Users may not understand formal notations like ER, DFD and UML diagrams.
- Some professional analysts develop informal models based on ad hoc diagrams, rich pictures, animations, PowerPoint simulations, text-based explanations and simple use cases for explaining requirements to users/clients.
- Informal models which are not based on formal notations like ER and UML are often the basis for agreement and sign-off for requirements specifications.

The implication for education and training from these ideas (and suggested by the study findings) is that for students to be able to undertake a major project they need to be encouraged to build their own personal toolkit after being exposed to as many tools and techniques as possible. This also implies that students should be encouraged to experiment with and develop some informal modelling techniques for communicating requirements to users/clients. There are many different modelling methods and notations available and that some are more appropriate for certain types of projects than others.

## 7 Conclusion

This paper has identified and discussed the implications of a set of case studies of professional requirements engineering practice regarding how professional requirements engineers reach agreement with clients on requirements specifications. The analysis of the findings is the basis of a theoretical model of the processes, concepts and relationships involved in the requirements engineering process with respect to the development and use of informal models for reaching understanding and agreement between users and analysts. The case study findings also suggest a need to examine education and training methods for requirements engineers and systems analysts with respect to developing diverse approaches to the use of models, methodologies, techniques and tools for elicitation, modelling and validation of user requirements.

## References

1. Avison D, Fitzgerald G (2003) *Information systems development: methodologies, techniques and tools*, 3rd edn. McGraw-Hill, Maidenhead
2. Bacharach S (1988) Organizational theories: some criteria for evaluation. *Acad Manag Rev* 14(4):496–515
3. Brown RBK, Piper IC (2011) What users do: SA&D with the ATSA method. Paper presented at the 20th international conference on information systems development (ISD2011), Edinburgh, UK
4. Carbone A, Lynch K, Arnott D, Jamieson P (2000) Adopting a studio-based education approach into information technology. Paper presented at the 4th Australasian computing education conference (ACE2000), Melbourne
5. Checkland P, Scholes J (1990) *Soft systems methodology in practice*. Wiley, Chichester
6. Conboy K (2010) Project failure en masse: a study of loose budgetary control in ISD projects. *Eur J Inform Syst* 19(3):273–287
7. Coughlan J, Macredie RD (2002) Effective communication in requirements elicitation: a comparison of methodologies. *Requirements Eng* 7(2):47–60
8. Darke P, Shanks G (1996) Stakeholder viewpoints in requirements definition: a framework for understanding viewpoint development approaches. *Requirements Eng* 1:88–105
9. Dawson L (2008) Active exploration of emerging themes in a study of object-oriented requirements engineering: the “Evolutionary Case” approach. *Electron J Bus Res Meth* 6(1):29–42
10. Dawson L (2011) Cognitive processes in object-oriented requirements engineering practice: analogical reasoning and mental modelling. Paper presented at the 20th international conference on information systems development (ISD2011), Edinburgh, UK
11. Dawson L, Darke P (2002) The adoption and adaptation of object-oriented methodologies in requirements engineering practice. Paper presented at the 10th European conference on information systems, Gdansk, Poland
12. Dawson L, Swatman P (1999) The use of object-oriented models in requirements engineering: a field study. Paper presented at the 20th international conference on information systems, Charlotte, NC
13. Dubin R (1976) Theory building in applied areas. In: Dunnette M (ed) *Handbook of industrial and organisational psychology*. Rand McNally College Publications, Chicago, IL, pp 17–39

14. Eisenhardt KM (1989) Building theories from case study research. *Academy of Management Review* 14(4):532–550
15. Fitzgerald B (1997) The use of systems development methodologies in practice: a field study. *Inform Syst J* 7:201–212
16. Flynn DJ, Warhurst R (1994) An empirical study of the validation process within requirements determination. *Inform Syst J* 4:185–212
17. Galal G, McDonnell JT (1998) A qualitative view of requirements engineering. Paper presented at the 3rd Australian conference on requirements engineering, Geelong, Australia
18. Goguen JA (1994) Requirements engineering as the reconciliation of social and technical issues. In: Jirotko M, Goguen JA (eds) *Requirements engineering: social and technical issues*. Academic, London
19. Jackson P, Klobas J (2008) Building knowledge in projects: a practical application of social constructivism to information systems development. *Int J Proj Manag* 26(4):329–337
20. Kavakli E (2002) Goal-oriented requirements engineering: a unifying framework. *Requirements Eng* 6(4):237–251
21. Keen CD, Lockwood C, Lamp J (1998) A client-focused, team-of-teams approach to software development projects. Paper presented at the software engineering: education and practice, Dunedin, New Zealand
22. Khushalani A, Smith R, Howard S (1994) What happens when designers don't play by the rules: towards a model of opportunistic behaviour and design. *Aust J Inform Syst* 1(2):2–31
23. Lamp J, Lockwood C (2000) Creating realistic experience of an IS project: the team of teams approach. Paper presented at the 2000 IRMA international conference, Alaska
24. Loucopoulos P, Karakostas V (1995) *Systems requirements engineering*. McGraw-Hill, London
25. Lyytinen K, Robey D (1999) Learning failure in information systems development. *Inform Syst J* 9(2):85–101
26. Macaulay L (1996) *Requirements engineering*. Springer, London
27. Machado R, Borges M, Gomes J (2008) Supporting the system requirements elicitation through collaborative observations. In: Briggs R, Antunes P, de Vreede G-J, Read A (eds) *Groupware: design, implementation and use*, vol 5411. Springer, Berlin, pp 364–379
28. Mathiassen L, Purao S (2002) Educating reflective systems developers. *Inform Syst J* 12(2): 81–102
29. Mayer RE (1992) *Thinking, problem solving, cognition*, 2nd edn. W.H. Freeman and Company, New York
30. Miles MB, Huberman AM (1994) *Qualitative data analysis: an expanded sourcebook*, 2nd edn. Sage Publications Inc., Thousand Oaks, CA
31. Nuseibeh B, Kramer J, Finkelstein A (1994) A framework for expressing the relationships between multiple views in requirements specification. *IEEE Trans Software Eng* 20:760–771
32. Pohl K (1994) The three dimensions of requirements engineering: a framework and its applications. *Inform Syst* 19(3):243–258
33. Robertson J, Robertson S (1997) *Volere requirements specification template*, 4th edn. Atlantic Systems Guild, London
34. Robertson J, Robertson S (2010) Volere product summary. <http://www.volere.co.uk/index.htm>. Accessed 6 Jul 2012
35. Schön DA (1983) *The reflective practitioner: how professionals think in action*. Basic Books, New York
36. Siau K, Rossi M (2011) Evaluation techniques for systems analysis and design modelling methods—a review and comparative analysis. *Inform Syst J* 21(3):249–268
37. Siau K, Wang Y (2007) Cognitive evaluation of information modeling methods. *Inform Software Technol* 49(5):455–474
38. Sommerville I, Sawyer P (1997) *Requirements engineering: a good practice guide*. Wiley, Chichester
39. Sutcliffe AG, Maiden NAM (1992) Analysing the novice analyst: cognitive models in software engineering. *Int J Man Mach Stud* 36:719–740

40. Teles VM, de Oliveira CET (2003) Reviewing the curriculum of software engineering undergraduate courses to incorporate communication and interpersonal skills teaching. Paper presented at the software engineering education and training, 2003 (CSEE&T 2003). Proceedings of 16th conference on 20–22 Mar 2003
41. Urquhart C (1998) Analysts and clients in conversation: cases in early requirements gathering. Paper presented at the 19th international conference on information systems, Helsinki, Finland
42. Wastell DG (1999) Learning dysfunctions in information systems development: overcoming the social defenses with transitional objects. *MIS Quart* 23(4):581–600