

Using Protocol Analysis to Explore the Creative Requirements Engineering Process

Lemai Nguyen

**Centre for Business Research, School of Information Systems,
Deakin University**

email: lemai@deakin.edu.au

Graeme Shanks

Faculty of Information Technology, Monash University

email: Graeme.Shanks@infotech.monash.edu.au

Abstract

Protocol analysis is an empirical method applied by researchers in cognitive psychology and behavioural analysis. Protocol analysis can be used to collect, document and analyse thought processes by an individual problem solver. In general, research subjects are asked to think aloud when performing a given task. Their verbal reports are transcribed and represent a sequence of their thoughts and cognitive activities. These verbal reports are analysed to identify relevant segments of cognitive behaviours by the research subjects. The analysis results may be cross-examined (or validated through retrospective interviews with the research subjects). This paper offers a critical analysis of this research method, its approaches to data collection and analysis, strengths and limitations, and discusses its use in information systems research. The aim is to explore the use of protocol analysis in studying the creative requirements engineering process.

Creativity in requirements engineering

Requirements engineering (RE), an early phase in information systems (IS) development, has been commonly agreed to be one of the most crucial phases in the development process (e.g. Boehm, 1981; Loucopoulos and Karakostas, 1995; Nuseibeh and Easterbrook, 2000). RE is concerned with the elicitation, modelling and specification of user requirements for the new system to be built. Recently, creativity has been increasingly seen as playing an important role in RE (Nguyen et al., 2000; Maiden and Gizikis, 2001; Robertson, 2005; Nguyen and Swatman, 2006; Maiden and Robertson, 2005).

Creativity involves the *exploration of conceptual spaces* by people in order to produce an outcome that is both *novel and useful* for a specific context (Boden, 1991; Plucker and Beghetto, 2004; Sternberg, 2005). Based on this understanding of creativity, we see two strong supporting arguments for the role of creativity in RE: creating a vision for ICT-enabled future business practice and developing a requirements specification for an information system to enable the vision. First, creating a vision into future ICT-enabled business practice is crucial in order to develop a new system with an objective to leverage the competitiveness of the organisation and effectiveness of its business functions (Robertson, 2002; Robertson, 2005). Robertson has advocated that the requirements process should involve a creative discovery of requirements to invent business processes rather than passively eliciting requirements from business users as currently described in the RE literature because ‘we won’t make significant improvements to our software products by following a logical train of thought’ (Robertson, 2005). Second, the RE process in its own right is not a purely deterministic, systematic process; it is an *exploration of conceptual spaces* involving cycles of structured and opportunistic insight-driven episodes (Nguyen et al., 2000; Nguyen and Swatman, 2003). Therefore, fostering and supporting creative thinking within the requirements gathering process is a key to effectively practice requirements engineering.

We argue that one major challenge in fostering and supporting creativity in RE is caused by the difficulty in obtaining a deep understanding of the creative cognitive process involved. For example, while all the practitioners participating in a focus group agreed that creativity was an essential requirement in all of their past requirements projects, they found it difficult to describe how the creative thinking process occurred (Cybulski et al., 2003). In their studies to select and integrate creativity techniques within RE, Maiden and Robertson (2005) criticised the fact that practitioners lack creativity theories and models to guide their creative process in RE. In response to this criticism, Nguyen and Shanks (2006) explored different facets of creativity, and especially different perspectives of creative processes in the creativity literature, and related them to creativity in RE. They concluded that an in-depth understanding of the creative cognitive process is required in order to effectively prompt and support creative thinking in RE.

In this paper, we explore the potential of protocol analysis, an empirical research method in cognitive psychology and behaviour analysis, for use in studying the creative RE process. The following section briefly describes different understandings of the problem solving process in RE and related fields to set a context for the following sections. After that, we describe and critically discuss the protocol analysis research method in terms of current approaches to data collection and analysis, and their benefits and limitations. Then we present a discussion of previous applications of protocol analysis in RE for different

research purposes with a view to assessing its relevance to research into the creative requirements process. Finally, we summarise the paper and outline future research directions.

Understanding the creative RE process

Characteristics of the RE process

We view the creative process in RE as having both emergent and design characterisations.

First, RE has been described as an ill-structured problem solving process. The ill-structuredness of the requirements problem can be characterised through the open-endedness of the problem, poorly understood problem context, existence of multiple domains, complexity and dynamics of social interactions, organisation structures, business processes and technologies involved (Guindon, 1990b; Conklin, 2005; Nguyen and Swatman, 2006). In ill-structured problem solving, the understanding (and discovery) of the problem and structuring of the solution are intertwined. The problem solver (i.e. the systems analyst) moves between different problem areas in search of a possible solution, interacts and communicates his or her understanding with other stakeholders, and responds to the emergent situation. Therefore, both the problem space and the solution space progressively evolve as the systems analyst gains more knowledge and responds to the stimuli produced by the social, business and technical complexity and dynamics. This is similar to a description of the creative design process: 'The designer operates within a context which partially depends on the designer's perceptions of purposes, constraints and related contexts. These perceptions change as the designer explores the emerging relationships between putative designs and the context and as the designer learns more about possible designs' (Gero, 1996).

Second, RE should be seen as an art involving two different acts — articulating and documenting user requirements (i.e. describing the real world situation) as well as designing new business practice (enabled by a new system) by suggesting changes to the current situation. These acts are referred to, respectively, as analysing and modelling As-Is and To-Be requirements. However, the literature tends to focus on the former more extensively. Requirements elicitation, modelling and communication are fundamental activities (Nuseibeh and Easterbrook, 2000; Loucopoulos and Karakostas, 1995; Pohl, 1994). The first activity focuses on the acquisition and articulation of the user requirements. The second activity focuses on the representation and documentation of the requirements in various formats and perspectives. The third activity aims at requirements communication, negotiation and validating a correspondence between the requirements specification and the real world problem. We acknowledge that these fundamental systematic activities are still required in

the development of To-Be requirements but we advocate that creative thinking plays a crucial role in envisaging and designing future information and communication technology (ICT) enabled business practice, especially if we aim at innovating in the business and creating new, significant added value through it. The invention of To-Be requirements, which is a largely missing activity in the current RE literature, is a key to envisaging and designing ICT enabled innovative business practice (Robertson, 2002, 2005). In addition, Simsion's (2006) investigation into data modelling in practice characterised data modelling as a creative design process although it is widely perceived to be a descriptive representation process.

Therefore, RE is a creative, emergent design process. In the next sub-section, we look at different views of the RE process and relate them to their counterparts in design studies and creativity research.

Views of the RE process

Nguyen and Swatman (2003, 2006) distinguished two views of the RE process that are held by members of the RE community:

The first view describes a systematic, structured and evolutionary process. Though detailed descriptions of the RE process may vary, essentially the requirements model is pictured as continually, incrementally structured and refined through a cyclic systematic process (e.g. Alexander, 1998; Loucopoulos and Karakostas, 1995; Kotonya and Sommerville, 1998).

The second view describes an opportunistic, constructivist process consisting of structuring and insight-driven restructuring of the requirements model. These opportunistic cognitive behaviours and insight-driven reconceptualisations of the problem space by the systems analyst are important in handling the emergent problem space and partial solutions (Guindon, 1990a; Visser, 1994; Khushalani et al., 1994; Nguyen et al., 2000; Robillard, 2005).

The existence of these two views of the RE process is reminiscent of the observation made by Dorst and Dijkhuis (1995) that there exist two views of the design process in the design studies community. The first view describes a rational problem solving process characterised by structured search and information processing in the problem space (Newell and Simon, 1972; Simon, 1969). The second view is constructivist and describes the design process as a reflective conversation with the situation (Schön, 1996).

Dorst and Dijkhuis (1995) have argued that the former characterisation describes the problem solving process for structured and fairly clear cut problems whereas the latter describes the conceptual design stage for ill-structured problems. Based on the discussion above, the latter matches the characteristics we attribute to the creative part of the RE process. We further note that these two descriptions of the design process are analogous to the two descriptions of the RE process.

We see this analogy as a manifestation of two ‘forces’ in RE: the enforcing of a systematic structured process to avoid a chaotic error-prone process, as opposed to the recognition and taking advantage of opportunistic cognitive behaviours and heuristics of professionals (in dealing with the emergent problem space). Both of these two forces are essential in RE problem solving; a good balance of them is required.

Boden (1991) has described the creative process as an internal process of exploration and transformation of conceptual spaces in an individual mind. However, understanding how this internal process, which actually happens in the individual mind, has long been a challenging topic in creativity research. There exist numerous models to describe the creative process. Shneiderman (2000) described three creative process models: inspirationalist, structuralist and situationalist.

The inspirationalist views the creative process as passing through four phases of preparation, incubation, illumination and verification, as in the model of Wallas (1926), and as unpredictable insight and associated restructuring of the problem space in Gestalt psychology (Mayer, 1992; Ohlsson, 1984). Common creativity enhancing techniques, such as lateral thinking, divergent thinking, six thinking hats, and free association, are often adopted to promote inspirational creativity.

The structuralist views the creative process as a more focused and structured effort to generate and evaluate ideas as in, for example, Osborn-Parnes’ Creative Problem Solving CPS (Daupert, 2002; Osborn, 1979) and the Directed Creativity Cycle (Plsek, 1997). A cyclic process of divergent brainstorming and convergent thinking phases is included in these models to stress a balance between imagination and the analytical aspects of creativity.

The situationalist emphasises the social interactions between individual problem solvers and thus stresses the collaborative nature of the creative process. Three out of the four phases of collecting, relating, creating and disseminating in the creative process model of Shneiderman (2000) are designed to directly support collaboration and communication of information and ideas in the creative process. In this vein, a group of researchers at the University of South Australia extended CSCW (Computer Support Collaborative Work) theories in order to develop ICT-enabled supportive workplace for creative teams (Blackburn et al., 2005).

We have two observations. First, the inspirationalist and the structuralist tend to focus on the creative effort by individuals while the situationalist emphasises the collaboration between them. Second, the inspirationalist tends to focus on how the creative process actually occurs while the structuralist and situationalist tend to focus on how the creative process should be undertaken. Based on a synthesis of creativity models from creativity research and creativity research in RE, Nguyen and Shanks (2006) suggested integrating the different views

through a collaborative process consisting of cycles of structured building up and opportunistic restructuring of the requirements model. This process needs to be further developed and refined. Overall, we conclude that the differences between different descriptions of the creative process that exist in the RE, design studies and creativity research communities manifest different styles of creative thinking and cognition. Our conclusion points to the need for, and difficulties in, integrating the different creative process views. It is therefore important to further explore creative thinking and cognition in the RE process and the question arises as to which research method(s) would be most appropriate to pursue this exploration.

To explore creative thinking and cognition, it is important to obtain data about the process and to reconstruct what happens in the mind of systems analyst. As highlighted earlier, a major challenge is that systems analysts can not describe accurately how they developed solutions for problems they faced (Lubars et al., 1993; Hofmann and Lehner, 2001). A similar challenge exists in design studies; designers 'cannot articulate what kind of expertise they use in designing and how' (Suwa et al., 1998). Researchers in design studies have been using protocol analysis, an empirical method in cognitive psychology, to examine the design process. In the next section, we describe and discuss the potential use of protocol analysis in exploring the creative process in RE.

Protocol analysis

Overview of protocol analysis

Researchers in the psychology of problem solving and design studies have recognised the importance of describing and understanding the cognitive process used by the problem solver. The belief is that a good understanding of the cognitive process would be useful to support and improve the problem solving and design practice and to effectively train practitioners. Protocol analysis is an empirical research method for studying the cognitive behaviours and thought processes used by problem solvers (Ericsson and Simon, 1993).

Protocol analysis usually takes place in a controlled environment. The research subject is a problem solver who is given a specific task and works individually on that task. Protocol analysis aims to collect as much detail as possible about the problem solving process, analysing the collected data and reconstructing what happens in the mind of the problem solver. On one hand, the controlled environment reduces 'noise' and allows the researcher to collect rich details and relevant data about the problem solving activities and artifacts produced during the experiment. On the other hand, shortcomings of this research method include a limited time period, a small problem, and the exclusion of social processes, teamwork and communication that often take place in everyday work.

There are different approaches to conducting protocol analysis. We discuss them in terms of data collection and data analysis.

Data collection approaches

There are two approaches to data collection in protocol analysis: concurrent and retrospective (Dorst and Dijkhuis, 1995; Ericsson and Simon, 1993).

Concurrent protocol

Concurrent protocols are generated when the problem solver verbalises their thoughts while working on a specific task. First, the problem solver is trained to verbalise his or her thoughts using a thinking aloud technique. Second, with a given task, the problem solver verbalises his or her thoughts while working on a given task. The process is video and/or audio taped, and transcribed. As a result, a thinking aloud concurrent protocol acts as the generator of the data source, which is then later coded and analysed.

Two assumptions underlie the validity of the verbalisation of thoughts process in concurrent protocols. The first is that the problem solving process has a conversational characteristic. Schön (1996) described design as a reflective dialogue of the problem solver with the materials of a problem situation. In developing a design rationale tool, Kaplan (1990) viewed the design process as a conversation-oriented activity, being either a monologue by one designer or conversations between different designers. The second is that the verbalisation of thoughts during the problem solving process will not affect the process. Ericsson and Simon (1993) describe three levels of verbalisation ranging from direct verbalisation without special effort to communicate thoughts, minimal intermediate processing to explicate the thought contents, and verbalisation with an explanation of thoughts, ideas and motives. Having reviewed empirical studies using these levels of verbalisation, Ericsson and Simon (1993) concluded that concurrent verbalisation does not alter the structure of thought processes. There is a disagreement about this conclusion. Lloyd et al. (1995) were concerned with the validity of concurrent protocols because thinking aloud may interfere with the problem solving process and, consequently, concurrent protocols may be incomplete and not reveal true insights into the actual problem solving process. A common view shared by the design studies research community is that concurrent protocols reveal a sequence of cognitive events and information processing stored in short-term memory (STM), thus providing rich details and opportunities for analysis to gain insight into the cognitive behaviours by the problem solver.

Retrospective protocol

Retrospective protocols conduct interviews with the problem solver after the problem solving process, usually immediately. During the interview, the problem

solver is asked to recall his or her activities. Interviews are audio and/or video taped and transcribed. The generated retrospective protocols serve as data for later coding and analysis to reconstruct the problem solving process and gain insight into what happened during the process.

While both concurrent protocol and retrospective protocol approaches share a common position that collected data can be used to reconstruct the problem solving process, the latter is often seen as less intrusive to the process under observation (Lloyd et al., 1995). However, Ericsson and Simon (1993) have argued that, after the experiment session is complete, information processing details are no longer accessible from STM because they have been transmitted into Long Term Memory (LTM) from which it is harder to retrieve. Consequently, the reconstructed process based on a retrospective protocol may be incomplete and inaccurate. Retrospective protocols may not show the actual sequence of cognitive events, instead they may show a rationalised or theorised story of the problem solving process. To address this in design studies, Suwa et al. (1998) suggested videotaping the design experiment session and using the videotapes to assist the retrieval of the cognitive events stored in LTM after the experiment session. In addition, the contents (sketches and diagrams) can also be collected for analysis. Guindon (1990a) supplemented her concurrent protocols with retrospective interviews to obtain additional design rationale and to gain a deep understanding of the designer's cognitive behaviours and the design process.

Gero and Tang (2001) conducted an empirical study to examine similarities and differences between concurrent and retrospective protocols. They found that both types of protocol methods show a similar frequency of changes of design intentions and consistent structures of the design process. They also found that the number of segments in a retrospective protocol is larger than the number of segments in a concurrent one. They explain that, through a revision of sketches and rehearsed memory after the thinking aloud session, the retrospective protocol produced more details than the concurrent protocol (Gero and Tang, 2001). The authors concluded that concurrent and retrospective protocols lead to similar results and that the concurrent protocol is an efficient and applicable method in understanding the design process.

Kuusela and Pallab (2000) conducted a similar comparative study using an experiment set in a context of customer decision making. Although the problem solving contexts and coding methods in studies by Gero and Tang (2001) and Kuusela and Pallab (2000) are different, a common conclusion was reached, namely that both concurrent and retrospective protocols lead to consistent understandings of the problem solving process. In addition, Kuusela and Pallab (2000) suggest that concurrent protocols are more suitable for examining the process while retrospective protocols are more suitable for examining the

outcome. Their conclusions support the potential use of protocol analysis to gain insight into the problem solving process in RE.

There are nevertheless two weaknesses with both concurrent and retrospective protocols. One of these is the well-known Hawthorne effect since both of these data collection approaches involve observation of a research subject who knows they are being watched. Other research approaches such as, for example, case study, action research and ethnography, also share this limitation (Neuman, 2003). Another weakness of protocol analysis is the difficulty in recruiting and training participants who are willing, capable and motivated to provide meaningful protocols. Previous successful applications of protocol analysis in design research have addressed this issue by explaining to participants the significance of the research and providing training that facilitates thinking aloud and articulating 'on the fly' thoughts.

Data analysis approaches

Data generated using either concurrent or retrospective protocols are coded (segmented) for the analysis and identification of cognitive patterns. First, the data is coded into segments. Often a change in the problem solver's intention, or the contents of their thoughts, signals a new segment. Second, the problem solving process is reconstructed as a sequence of coded segments. Finally, correlations between segments are identified. Based on the two views of the design process, rational problem solving and constructivist, there are two approaches to segmenting data: process-oriented and content-oriented (Dorst and Dijkhuis, 1995; Gero and Neill, 1998).

Process-oriented segmentation

The process-oriented segmentation approach aims at describing the design process as a sequence of problem solving activities, using a problem solving taxonomy such as, for example, problem recognition, goal setting, solution proposing, solution analysing, or top down vs. bottom up strategies. In this approach, the protocol transcriptions are often coded into segments by syntactic markers, such as pauses, intensity, intonations, phrases and sentences that then aggregate into cognitive units called design intentions or design moves, for analysis (Ericsson and Simon, 1993). Alternatively, protocols can be directly segmented by design intentions based on the problem solving taxonomy — for example, problem domain including abstraction levels, functions, behaviours, structures; and micro and macro design activities such as proposing solutions, analysing solutions, explicit strategies, top down, bottom up, opportunistic (Gero and Neill, 1998). The categorisation of design intentions is often determined before the segmentation of the protocol. Gero and Neil (1998) also suggest open segmentation of protocols to allow new categories to emerge during the segmentation process. The segments generated from the protocol are often

quantitatively analysed to identify time spent on different types of design intentions, and to reconstruct a sequence of, and correlations between, them.

Benefits of process-oriented segmentation include: a design process described in the form of a sequence of design intentions and an understanding of correlations between design intentions, often presented in a graph form. Dorst and Cross's (2001) protocol analysis, involving an evaluation of nine creative designs in industrial design experiment, offered a refined model of a co-evolution of both the problem space and solution space. Their study supported Schön's (1983) argument that insight-driven problem (re)framing is crucial to the creative design process. Another example is a study by Guindon (1990a) involving eight designers in a lift control software design experiment. This study is often cited in the RE literature. Using a process-oriented segmentation method to examine concurrent protocols produced in this study, Guindon (1990a) observed significant deviations from a systematic structured process. She was amongst the first authors to propound opportunistic cognitive behaviours in high-level software design. Opportunistic behaviours and deviation from a structured process were also observed and reported in requirements engineering by Khushalani et al. (1994) and Nguyen et al. (2000).

Dorst and Dijkhuis (1995) have criticised the process-oriented approach on the basis that it fails to examine what designers see and think and what knowledge they exploit. This weakness can be addressed using the content-oriented segmentation approach.

Content-oriented segmentation

The content-oriented approach to protocol segmentation focuses on the cognition of the problem solver; that is, what he or she sees and thinks and what knowledge he or she uses (Suwa and Tversky, 1997; Suwa et al., 1998). There are two types of cognitive contents: visual contents (depicted elements and their spatial relations as drawn in the artifacts, and movements such as eye movement, moving pencils, etc) and non-visual contents (including thoughts and knowledge). A well defined classification of content-oriented segments (Tang and Gero, 2000) includes:

- Physical — depiction, looking, motion;
- Perceptual — perceiving depicted elements and their relations;
- Functional — assigning meaning to depictions/perception; and
- Conceptual — goal setting and decision making.

To study discontinuity and unexpected discoveries in the design process, design segments are indexed as being new, continual or revisited.

The content-oriented segmentation approach has been found to be useful in examining cognitive interactions between designer and artifacts. Using a

content-oriented segmentation classification scheme, Suwa et al. (1998) found that sketches can be seen as an external memory useful for subsequent inspections, visual cues for functional actions, and a physical setting for functional thoughts to be constructed on the fly in the emergent problem situation. The use of sketches was also investigated in a recent study (Bilda et al., 2006) using a revised content-oriented segmentation scheme. This study found that sketching or externalising may be useful but not necessary to design in terms of developing a network of ideas, pursuing cognitive activities and obtaining a satisfactory outcome. As systems analysts often use requirements models to represent and communicate requirements with each other and with other stakeholders, interactions between systems analysts and requirements models can be examined using content-oriented protocol analysis.

According to Tang and Gero (2000), there are two types of content-oriented segments and both are essential in the design process. Goal-driven segments reflect the rational problem solving process (Newell and Simon, 1972) and sensor-driven segments reflect the constructivist and reflection-in-action process (Schön, 1983). To us, this observation can be related to the description of catastrophe cycles in the requirements gathering process (Nguyen et al., 2000; Nguyen and Swatman, 2003).

In summary, the content-oriented and process-oriented segmentation approaches can both be beneficial. In RE, the invention or discovery of requirements and changes to requirements models should be studied in relation to associated cognitive behaviours to evaluate the creative requirements process and their impact on the creative outcome. There are, though, two common weaknesses from the point of view of RE in current segmentation classification schemes. First, both process-oriented and content-oriented segmentation approaches need to be adjusted to the RE knowledge domain, tailored, for example, to a particular requirements method and process. Second, segment classification should be linked to different types of creativity and creative thinking styles such as, for example, exploratory, combinatorial, analogy, transformation, structured and unstructured (Boden, 1991; Ward and Finke, 1999; Sternberg, 2005).

Discussion

Protocol analysis is widely used in problem solving research, especially in design studies. As the debate about the strengths and weaknesses of protocol analysis continues, this research method evolves. In terms of data collection protocols, comparative studies tend to confirm that concurrent and retrospective protocols produce similar results (Tang and Gero, 2000; Kussela and Pallab, 2000). In terms of data segmentation and coding, segmentation schemes are developed to enable researchers to gain in-depth understandings of the process as well as the interaction between the designer and artifacts (Gero and Neill, 1998; Tang and Gero, 2000; Bilda et al., 2006).

Protocol analysis has also been adopted and adapted to studying thinking processes in teams. For example, Stempfle and Badke-Schaub (2002) recorded team concurrent communication and analysed the generated protocol sentence by sentence. They developed a new coding scheme to examine collective design actions. Amongst others, important findings concerned the structuring of group process and, a continual 'interweaving of content-oriented and process-oriented sequences', and a tendency to immediately evaluate new ideas by team members (Stempfle and Badke-Schaub, 2002). In our view, since the pseudo-concurrent protocol did not capture verbalised thoughts, the retrospective protocol may be complementary: a combination of intermediate artifacts, video tapes and retrospective interviews can be useful in reconstructing multiple cognitive processes and teamwork dynamics. Distributed cognition theories can be also adopted to investigate creative team processes.

It is interesting to observe that, between the 1990s and early 2000s, design studies and RE researchers have come up with similar observations about the creative, emergent problem solving process and the co-evolution of the problem space and the solution space. But researchers in design studies have used protocol analysis, and proactively invented new segmentation schemes to examine the creative design process while RE researchers have used other research approaches, as will be discussed below.

Discussion and conclusion

Applications of protocol analysis in requirements engineering

Protocol analysis has been applied to the study of the cognitive behaviours of software and database designers (Guindon, 1990a; Sutcliffe and Maiden, 1992) and systems analysts (Batra and Davis, 1992; Chaiyasut and Shanks, 1994). A majority of these studies focus on categorising cognitive behaviours exhibited by systems designers or analysts and/or examining similarities and differences between novices and experts.

Guindon (1990a) discovered that the ill-structuredness of the requirements problem was an important factor inducing the opportunistic behaviours of the software designer. The opportunistic behaviours are associated with inferences related to new, emergent details associated with the incompleteness and ambiguity of the ill-structured problem. Often, upon sudden discovery of such details, a designer tends to immediately develop new partial solutions and test and modify them, rather than continuing to work on their previous planned task at a higher abstraction level. Their traversal between different abstraction levels was not systematic.

Sutcliffe and Maiden (1992) analysed verbal protocols supplemented by retrospective questionnaires from the development of a requirements specification

for a delivery scheduling system. They were able to categorise and model the cognitive behaviours as consisting of complex dependencies between information-gathering, assertions, conceptual modelling, planning, recognising goals and reasoning. According to these authors, the strongest associations were between information-gathering, assertions and conceptual modelling. These associations were explained as a representation of the analytical side of understanding the problem domain.

Batra and Davis (1992) examined similarities and differences between novice and expert database designers and concluded that novices focused on structuring requirements while experts' efforts were directed towards developing a holistic understanding of the problem, abstracting, categorising and representing. They noted cyclic movements between problem understanding and problem modelling by experts. With a focus on the data aspect, Chaiyasut and Shanks (1994) examined differences between data models produced by expert and novice data modellers. The authors categorised the cognitive process into six detailed types and noted that novices' models were developed 'literally' from the problem description while experts' models were more comprehensive, complete and held a holistic view of the problem.

Other studies using protocol analysis in requirements research are not related to the creative requirements process. For instance, protocol analysis was adopted as a research method in evaluating conceptual tools in modelling composites, data and properties (Shanks et al., 2003). More recently, Owen and his colleagues (Owen et al., 2006) criticised a lack of applications of protocol analysis in software engineering research and demonstrated benefits of protocol analysis as a research method in gaining valuable insight into how human factors influenced the interpretation and use of technical documentation by systems developers. Interestingly, protocol analysis was not only seen as a research method but also suggested as a way to observe and learn about the requirements problem context through users' work patterns and behaviours. Protocol analysis was also included as a technique in the ACRE framework, a framework to guide practitioners in requirements acquisition (Maiden and Rugg, 1996).

Can protocol analysis be used to study creative thinking and cognition in the requirements process?

We have reviewed various research approaches to studying the creative RE process and behaviours of the systems analysts. The table below summarises our findings in terms of their strengths and weaknesses.

Research method	References	Strengths	Weaknesses
Laboratory experiment	(Khushalani et al., 1994; Simson, 2006)	More control over the process Gains insight into the process and outcome by individual in small tasks	Limited time and small tasks Difficult to study collective creative problem solving process
Survey	(Simson, 2006)	Investigates specific well-defined constructs and concepts	Difficult to explore new concepts and gain in-depth understandings of why and how Difficult to reconstruct non-verbal thinking processes and cognitive activities
Protocol analysis	(Guindon, 1990a; Sutcliffe and Maiden, 1992; Batra and Davis, 1992; Chaiyasut and Shanks, 1994)	Can be designed to have a more natural setting compared to lab experiments, similar to workshops. Generates rich data to gain insight into non observable thinking process by individual problem solvers	Often limited time and small tasks Difficult to study situational collaborative creative process Difficulty to recruit and train participants
Workshop observation/ Positivist case study	(Khushalani et al., 1994; Maiden and Robertson, 2005)	More control of procedures and tasks, less control of interactions and group dynamics May gain access to the situational collaborative process Useful to confirm or disconfirm hypotheses and explore and identify issues for further studies	Difficult to reconstruct non-verbal thinking process and cognitive activities Less control over the process, difficult to find host Contextual, often limited time
Focus group	(Cybulski et al., 2003)	Good to explore/validate multiple view points through panel interviews	Difficult to reconstruct non-verbal thinking process and cognitive activities Group dynamics may be interfere with results, contextual
Interpretive case study/ Action research	(Nguyen et al., 2000; Dallman et al., 2005; Raisey et al., 2005)	May gain access to the situational collaborative process Data-grounded and inductive Improving practice (action research)	Difficult to reconstruct non-verbal thinking process and cognitive activities Less control over the process, difficult to find host Contextual

Table 1: Existing research approaches to studying the systems analysts' behaviours in RE.

As we have argued, cognitive behaviours are important in studying creative processes in RE. In obtaining data about the cognitive behaviours, a challenge faced by researchers is that creative thinking involves both verbal and non-verbal activities. While verbal activities (meetings, conversations, requirements

workshops, and group brainstorming) can be observed directly to generate data, non-verbal activities (silent cognitive behaviours that occur in the mind of the systems analyst) are much harder to access. Therefore, protocol analysis can be a key research method to gain insight into cognition and creative thinking in the requirements process. Surprisingly, protocol analysis was used to study cognitive behaviours in RE in the early 1990s although it has not been used specifically in studying the creative requirements process.

We suggest two possible applications of protocol analysis: using it to examine creative thinking and cognition in the creative requirements process; and using it to evaluate different requirements processes that utilise creativity techniques (such as that of Maiden and Robertson, 2005) and that do not utilise creativity techniques (for example, UML in Dennis et al., 2002) in relation to assessing the creative outcome produced through using these different RE processes.

Protocol analysis comes with inherent limitations: limited generalisability to real commercial projects and weak suitability to study collaborative process. To address these, a combination of research methods can be useful. In fact, a number of authors adopt a combination of different research approaches. For example, Simson (2006) used interviews, surveys and laboratory experiments in studying creativity in data modelling, and Khushalani et al. (1994) used workshop observations and laboratory experiments to examine opportunistic behaviours by systems designers. Protocol analysis can, we argue, potentially be used with other fieldwork research approaches and to study cognitive behaviours in the creative, emergent and collaborative process in RE in particular.

References

- Alexander, I. 1998, 'Requirements engineering as a co-operative inquiry: A framework', Proceedings of the Conference on European Industrial Requirements Engineering CEIRE '98, London, UK.
- Batra, D. and Davis, J. G. 1992, 'Conceptual data modelling in database design: Similarities and differences between expert and novice designers', *International Journal of Man-Machine Studies*, vol. 37, pp. 83-101.
- Bilda, Z., Gero, J. S. and Purcell, T. 2006, 'To sketch or not to sketch? That is the question', *Design Studies*, in Press.
- Blackburn, T., Swatman, P. and Vernik, R. 2005, 'Extending CSCW theories to model and support creative group processes', InSyL Working Papers, School of Computer and Information Science, University of South Australia.
- Boden, M. A. 1991, *The Creative Mind: Myths and Mechanisms*, Basic Books.
- Boehm, B. W. 1981, *Software Engineering Economics*, Englewood Cliffs, Prentice Hall.

- Chaiyasut, P. and Shanks, G. 1994, 'Conceptual data modelling process: A study of novice and expert data modellers', Proceedings of the 1st International Conference on Object-Role Modelling, Magnetic Island, Queensland, Australia.
- Conklin, J. 2005, *Dialogue Mapping: Building Shared Understanding of Wicked Problems*, John Wiley and Sons.
- Cybulski, J., Nguyen, L., Thanasankit, T. and Lichtenstein, S. 2003, 'understanding problem solving in requirements engineering: debating creativity with IS practitioners', Proceedings of the 7th Pacific Asia Conference on Information Systems, PACIS'2003, Adelaide, Australia.
- Dallman, S., Nguyen, L., Lamp, J. and Cybulski, J. 2005, 'Contextual factors which influence creativity in requirements engineering', Proceedings of 13th European Conference on Information Systems ECIS 2005, Regensburg, Germany.
- Daupert, D. 2002, *The Osborne-Parnes Creative Problem Solving Process Manual*.
- Dennis, A., Wixom, B. H. and Tegarden, D. 2002, *Systems Analysis and Design An Object-Oriented Approach with UML*, John Wiley and Sons.
- Dorst, K. and Cross, N. 2001, 'Creativity in the design process: Co-evolution of problem—solution', *Design Studies*, vol. 22, pp. 425—37.
- Dorst, K. and Dijkhuis, J. 1995, 'Comparing paradigms for describing design activity', *Design Studies*, vol. 16, pp. 261-74.
- Ericsson, K. A. and Simon, H. A. 1993, *Protocol Analysis Verbal Reports as Data*, The MIT Press, Cambridge, Massachusetts.
- Gero, J. S. 1996, 'Creativity, emergence and evolution in design: Concepts and framework', *Knowledge-Based Systems*, vol. 9, no. 7, pp. 435-48.
- Gero, J. S. and Neill, T. M. 1998, 'An approach to the analysis of design protocols', *Design Studies*, vol. 19, pp. 21-61.
- Gero, J. S. and Tang, H.-H. 2001, 'The differences between retrospective and concurrent protocols in revealing the process-oriented aspects of the design process', *Design Studies*, vol. 21, no. 3, pp. 283-95.
- Guindon, R. 1990a, 'Designing the design process: Exploiting opportunistic thoughts', *Human-Computer Interaction*, vol. 5, pp. 305-44.
- Guindon, R. 1990b, 'Knowledge exploited by experts during software system design', *International Journal of Man-Machine Studies*, vol. 33, pp. 279-304.
- Hofmann, H. F. and Lehner, F. 2001, 'Requirements engineering as a success factor in software projects', *IEEE Software*, vol. 18, no. 4 pp. 58-66.

- Kaplan, S. M. 1990, 'COED: A conversation-oriented tool for coordinating design work', in Finkelstein, A., Tauber, M. J. and Traummuller, R. (eds), *Human Factors in Information Systems Analysis and Design*, North-Holland, Elsevier Science Publishers, pp. 123-42.
- Khushalani, A., Smith, R. and Howard, S. 1994, 'What happens when designers don't play by the rules: Towards a model of opportunistic behaviour in design', *Australian Journal of Information Systems*, vol. 1, no. 2, pp. 13-31.
- Kotonya, G. and Sommerville, I. 1998, *Requirements Engineering: Processes and Techniques*, West Sussex, England, John Wiley and Sons.
- Kuusela, H. and Pallab, P. 2000, 'A comparison of concurrent and retrospective verbal protocol analysis', *American Journal of Psychology*, vol. 113, no. 3, pp. 387-404.
- Lloyd, P., Lawson, B. and Scott, P. 1995, 'Can concurrent verbalisation reveal design cognition?', *Design Studies*, vol. 16, pp. 237-59.
- Loucopoulos, P. and Karakostas, V. 1995, *System Requirements Engineering*, New York McGraw-Hill.
- Lubars, M., Potts, C. and Richter, C. 1993, 'A review of the state of the practice in requirements modelling', *Proceedings of the 1st IEEE International Symposium on Requirements Engineering RE'93*, San Diego, California, IEEE Computer Society Press.
- Maiden, N. and Gizikis, A. 2001, 'Where do requirements come from?', *IEEE Software*, vol. 18, no. 5, pp. 10-2.
- Maiden, N. and Robertson, S. 2005, 'Integrating creativity into requirements engineering process: Experiences with an air traffic management system', *Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05)*, Paris.
- Maiden, N. and Rugg, G. 1996, 'ACRE: Selecting methods for requirements acquisition', *Software Engineering Journal*, vol. 11, no. 3, pp. 183-92.
- Mayer, R. E. 1992, *Thinking, Problem solving, Cognition*, New York, W. H. Freeman and Company.
- Neuman, W. L. 2003, *Social Research Methods: Qualitative and Quantitative Approaches*, Allyn and Bacon.
- Newell, A. and Simon, H. A. 1972, *Human Problem Solving*, Englewood, Prentice-Hall.
- Nguyen, L., Carroll, J. and Swatman, P. A. 2000, 'Supporting and monitoring the creativity of IS personnel during the requirements engineering

- process', Proceedings of the 33rd Hawai'i International Conference on System Sciences, HICSS-33, Maui, Hawaii.
- Nguyen, L. and Shanks, G. 2006, 'Exploring creativity elements in requirements engineering', Forthcoming.
- Nguyen, L. and Swatman, P. A. 2003, 'Managing the requirements engineering process', *Requirements Engineering*, vol. 8, no. 1, pp. 55-68.
- Nguyen, L. and Swatman, P. A. 2006, 'Promoting and supporting requirements engineering creativity', in Dutoit, A. H., McCall, R., Mistrik, I. and Paech, B. (eds), *Rationale Management in Software Engineering*, Springer-Verlag.
- Nuseibeh, B. A. and Easterbrook, S. M. 2000, *Requirements Engineering: A Roadmap*, IEEE Computer Society Press.
- Ohlsson, S. 1984, 'I. Restructuring revisited: Summary and critique of the gestalt theory of problem solving', *Scandinavian Journal of Psychology*, vol. 25, pp. 65-78.
- Osborn, A. F. 1979, *Applied Imagination: Principles and Procedures of Creative Problem-Solving*, New York, Charles Scribner's Sons.
- Owen, S., Budgen, D. and Brereton, P. 2006, 'Protocol analysis: A neglected practice', *Communications of the ACM*, vol. 49, no. 2, pp. 117-22.
- Plsek, P. E. 1997, *Creativity, Innovation, and Quality*, Quality Press
- Plucker, J. A. and Beghetto, R. A. 2004, 'Why creativity is domain general, why it looks domain specific, and why the distinction does not matter', in Sternberg, R. J., Grigorenko, E. G. and Singer, J. L. (eds), *Creativity: From Potential to Realisation*, American Psychological Association (APA).
- Pohl, K. 1994, 'Three dimensions of requirements engineering: A framework and its application', *Information Systems*, vol. 19, no. 3, pp. 243-58.
- Raisey, D., Tan, K., Swatman, P., Blackburn, T. and Nguyen, V. 2005, 'An empirical study of the evolving dynamics of creative teams in action', InSyl Working Papers, School of Computer and Information Science, University of South Australia.
- Robertson, J. 2002, 'Eureka! Why analysts should invent requirements', *IEEE Software*, vol. 19, no. 4, pp. 20-2.
- Robertson, J. 2005, 'Requirements analysts must also be inventors', *IEEE Software*, vol. 22, no. 1, pp. 48, 50.
- Robillard, P. N. 2005, 'Opportunistic problem solving in software engineering', *IEEE Software*, vol. 22, no. 6, pp. 60-7.

- Schön, D. A. 1983, *The Reflective Practitioner: How Professionals Think in Action*, London England, Temple Smith.
- Schön, D. A. 1996, 'Reflective conversation with materials', in Winograd, T. (ed.) *Bringing Design to Software*, New York, ACM Press, pp. 171-84.
- Shanks, G., Tansley, E. and Weber, R. 2003, 'Using ontology to help validate conceptual models', *Communications of the ACM*, vol.46, pp. 85-89.
- Shneiderman, B. 2000, 'Creating creativity: User Interfaces for supporting innovation', *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 7, no. 1, pp. 114-38.
- Simon, H. A. 1969, *The Sciences of the Artificial*, The MIT Press, Cambridge, Massachusetts.
- Simsion, G. C. 2006, 'Data modelling: Description or design?', unpublished PhD thesis, Department of Information Systems, University of Melbourne, Melbourne, Australia.
- Stempfle, J. and Badke-Schaub, P. 2002, 'Thinking in design teams — an analysis of team communication', *Design Studies*, vol. 23, pp. 473-96.
- Sternberg, R. J. 2005, 'Creativity or creativities?', *International Journal of Human-Computer Studies*, vol. 63, pp. 370-82.
- Sutcliffe, A. G. and Maiden, N. A. M. 1992, 'Analysing the novice analyst: Cognitive models in software engineering', *International Journal Man-Machine Studies*, vol. 36, pp. 719-40.
- Suwa, M., Purcell, T. and Gero, J. 1998, 'Macroscopic analysis of design processes based on a scheme for coding designers' cognitive actions', *Design Studies*, vol. 19, pp. 455-83.
- Suwa, M. and Tversky, B. 1997, 'What do architects and perceive in their design sketches? A protocol analysis', *Design Studies*, vol. 18, pp. 385-403.
- Tang, H.-H. and Gero, J. S. 2000, 'Content-oriented coding scheme for protocol analysis and computer-aided architectural design', *Proceedings of the 5th Conference on Computer Aided Architectural Design Research in Asia, CAADRIA2000*, Singapore.
- Visser, W. 1994, 'Organisation of design activities: Opportunistic, with hierarchical episodes', *Interacting with Computers*, vol. 6, no. 3, pp. 235-74.
- Wallas, G. 1926, *The Art of Thought*, London England, Jonathan Cape.
- Ward, T. B. and Finke, R. A. 1999, 'Creative cognition', in Sternberg, R. J. (ed.), *Handbook of Creativity*, New York, Cambridge University Press, pp. 189-212.