Using Expertise as a Framework for Evaluating Requirements Technology

Ban Al-Ani and Susan Elliott Sim
Donald Bren School of Information and Computer Sciences, University of California, Irvine
{balani|sesim}@uci.edu

Abstract

The field of requirements engineering (RE) is rich with a myriad of technologies that aim to support activities within the requirements engineering process. These range from elicitation to management of RE work products. As such, the range and nature of these technologies are broad; this makes an evaluation of their effectiveness challenging. This position paper proposes an expertise framework as a basis for conducting comparative evaluation of RE technologies. Expertise is characterized using three dimensions: Practical Skill, Theoretical Knowledge, and Problem Domain. Expertise is increased by applying different processes to each dimension: Practical Skill needs to be abstracted; Theoretical Knowledge needs to be conditionalized, and Problem Domains need to be Contextualized. These dimensions and processes can be used as the basis for evaluating requirements tools and methods. In other words, to what extent does a particular technology raise the level of expertise for a requirements engineer by augmenting one of these processes? The framework is presented by applying it to evaluate four different techniques (ethnography, use cases, scenarios, and traceability matrix).

1. Introduction

The aim of requirements engineering (RE) research is to produce technologies to help professional software developers elicit, specify, analyse, validate, communicate, and manage requirements more effectively. These technologies can take the form of tools, methods, techniques, notations, or even education. The broad range of activities and variety of technologies makes it very challenging to use a uniform basis for evaluation. Though it is possible to measure how much technology has contributed when it replaces humans, it is notoriously difficult to measure how much computers assist humans [7].

We propose to use expertise as a basis for the evaluation of requirements technology. A tool or method that is helpful enables a requirements engineer to perform more effectively, that is, to perform a job more expertly. In this paper, we present this expertise-based framework and demonstrate how it can be used to conduct comparative evaluations.

This position paper presents the findings of ongoing research in the following sections. The first section details the meaning of expertise as perceived by this research. The proposed framework is presented and then implemented to appraise four widely utilised technology from different phases of RE. Finally, the paper outlines future work in the concluding remarks.

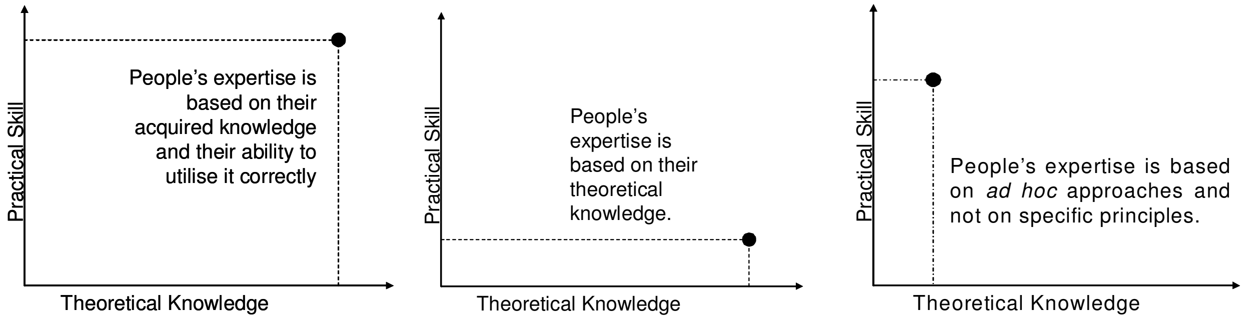
2. Expertise

An expert is someone who consistently performs at a high level in a specific field of human activity [13]. This skill is often accompanied by a track record of accomplishments and consequently the achievements cannot be attributed to luck. It typically takes 10,000-20,000 hours of deliberate practice to become an expert [3].

It is widely accepted that knowledge is only useful when correctly applied. The depth of knowledge and the skill to utilise it correctly can determine the level of expertise. This leads to the conclusion that expertise is a complex function of theoretical knowledge and practical skills that will determine an individual's performance in a particular domain. Consequently, this definition of expertise can be represented using a mathematical notation (Definition 1) and graphically (Figure 1):

Definition 1: The relationship between performance, expertise, knowledge, skill and domain.

Performance = Expertise (knowledge, skill, domain)



1.a. A person with high level of practical skills and high level of theoretical knowledge.

1.b. A person with small degree practical skills and high level of theoretical knowledge.

1.c. A person with high level of practical skills and high of theoretical knowledge.

Figure 1. A representation of expertise as a combination of practical skill and theoretical knowledge.

The term *knowledge* refers to awareness, or understanding gained through study and is therefore considered *theoretical knowledge*. The term *skill* refers to a proficiency, facility, or dexterity that is acquired or developed through training or experience. It is therefore considered *practical skills* in the context of this research. Finally, the term *domain* refers to the *problem domain*, which is the world where the problem is located and the quality of its solution will be evaluated [6].

Figure 1.a, represents a requirements engineer that possesses knowledge of the activities within the Requirements Engineering (RE) process, for example, and the ability (skill) to carry out those activities. Someone who has read a RE textbook would have knowledge of the basic underlying principles but have limited (if any) skill in applying that knowledge. This is represented in Figure 1.b. Another extreme possibility is someone who has developed their RE expertise based on trial and error. Such a person possesses the skill necessary to carry out RE activities but will typically do so in an *ad hoc* manner and is represented in Figure 1.c.

3. RE Expertise Framework

In previous work we proposed a model of RE expertise in which the areas of RE expertise consists of expertise in elicitation, analysis, communication, validation and management. A requirements engineer can be categorised as being naïve, novice, beginner, professional, expert or an elite requirements engineer based on their expertise [1]. Some of the variables that were considered key to determining expertise in requirements engineering are as follows: education, years of practice, nature/quality of practice, domain of practice, and so on.

Our initial characterisation of expertise is extended using Dickenson's three point categorisation of expertise definitions [2]:

1. Experts approach problems in **abstract** ways, which are not obvious or perhaps even comprehensible to a novice. Experts are able to do this because their grasp of conceptual models enables them to 'see' relationships and patterns between seemingly unrelated ideas, data and facts.
2. Experts are able to conditionalise their knowledge to develop some form of context. Experts only utilise knowledge when the context is correct. This **conditional knowledge**, by definition, is subject to change. Experts are able to rapidly assimilate new knowledge, assign it relevant context and alter their existing conceptual models to take account of this new information.
3. Information in and of itself has no utility. It is only when information is assigned relevance, **context** or is associated with other information does it hold any value. If this information can maintain this meta-data of its own utility, even when removed from the structure that assigned it the information, then the information retains value.

Dickenson provides only this categorization; the remaining work is ours, in particular, the proposed framework illustrated in Figure 2.

The foundation of the proposed framework is based on the three principal joists, namely: practical skills, theoretical knowledge, and the problem domain. Various studies contribute to this representation of expertise and also indicate that joist precincts are not distinctly defined (e.g. [8]). The area within these three points denotes the extent of expertise. While other characterisations of expertise can exist, we elected to investigate these in our initial study.

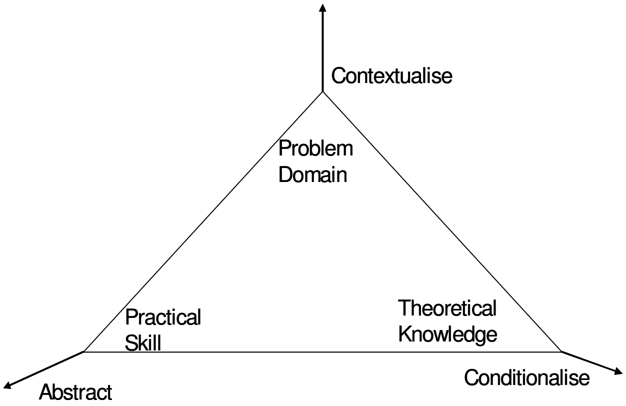


Figure 2. An Expertise framework which illustrates the relationship between requirements engineers' practical skill and theoretical knowledge within a specific domain and their ability to abstract, conditionalise, and contextualise.

The framework also illustrates that a requirements engineers ability to abstract existing practical skills, adapt theoretical knowledge conditionally and contextualise both skill and knowledge within a specific problem domain will increase their expertise. Consequently, a technology that increases a requirements engineer's ability to contextualise, conditionalise and abstract existing knowledge and skill within a problem domain suggests that is an effective technology.

This framework can provide a useful way of performing comparative analytical evaluation because it is technology agnostic. It can be applied to tools and techniques from different stages of the requirements process. In addition, being an analytical or conceptual framework, it provides a common basis for combining or comparing results from different evaluations through meta-analysis.

The vectors or processes in the model have interesting analogues to the different ways that research results are captured and published. Practical skills that have been abstracted are typically published as an experience report. Theoretical knowledge often comes from inventing new requirements technologies, but the knowledge and technology only becomes conditionalised when we learn when and where to apply them. Finally, the problem domain becomes contextualized when we have domain-specific requirements technologies or re-usable requirements for a domain.

In this paper we argue that a requirements engineering technology can be said to be of greater effectiveness if it raises the users' performance, i.e. moves him or her to a more proficient expertise level. The technology does so by allowing a requirements engineer to abstract, contextualise, and conditionalise knowledge.

4. Applying Proposed Framework

We illustrate the application of the framework by applying it to well-known requirements technologies. We selected four technologies to represent different levels of formality and a range of requirements engineering phases. The results of conducting a comparative analysis between the characteristics of these technologies with the proposed framework is summarised in table 1.

The findings presented in table 1 lead to the conclusion that not all requirements technologies are equally effective and that their effectiveness does not rely solely on the problem domain but also on the technology's ability to increase its users' expertise.

For example, ethnography does not support abstraction or a rapid change of a conceptual model. While it might be an effective technology to adopt by requirements engineers with a high level of expertise it should be treated with caution when considered by a novice. Creating and maintaining a traceability matrix, however, can lead to an increase in expertise within a particular problem domain.

5. Concluding Remarks

Two principal definitions were presented in this paper, namely:

1. **Expertise:** expertise was defined as a complex function of knowledge and skill that will determine an individual's performance in a particular problem domain. The proposed definition provided a foundation for the proposed expertise framework.
2. **Expertise Framework:** We combined the expertise definition and the three point categorisation scheme defined by Dickenson [2] to create a novel expertise framework that can be utilised to evaluate RE technology's effectiveness. A summary of evaluation findings was tabulated and also presented in this paper.

Research is ongoing to investigate the meaning and characteristics of expertise within requirements engineering. While we have defined areas of RE expertise, we have not identified a means to determine the level of expertise achieved by an individual. For example, what are the indicators demonstrated by a *professional* requirements engineer? How is expertise manifested? Acquired? Measured? Answers to these questions can assist in more appropriate allocation of requirements engineers to task (increase productivity), providing a level of support for an RE technology that is appropriate to anticipated requirements engineers' expertise and developing more RE courses that will equip students with an appropriate level of expertise.

Future research includes conducting the following:

1. Refinement of and validation of the proposed framework through empirical studies. We plan to involve both industry and academic practitioners.
2. Development of metrics to measure an individual's skill and knowledge within a particular domain.

6. Acknowledgements

This work was supported in part by the National Science Foundation under award number CCF-0430026. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

7. References

[1] Al-Ani, B and Sim, S., (2006). "So, You Think You're A Requirements Engineer?", Extended abstract, Proceedings 14th IEEE International Conference on Requirements Engineering, Minneapolis/Minnesota 2006, forthcoming.

[2] Dickinson, Guy S (2003). "Weblogs - can they accelerate expertise?", MA Education thesis, Ultralab, Anglia Polytechnic University, Manchester, UK.

[3] Ericsson, K. A. and Charness, N., "Expert Performance: Its Structure and Acquisition," American Psychologist, 49, 8, August, (1994), pp. 725-747.

[4] Gotel, O.C.Z. and Finkelstein, C.W.; (1994). "An analysis of the requirements traceability problem", Proceedings of the First International Conference on Requirements Engineering, 1994., 18-22 April 1994 pp. 94 - 101

[5] Hughes, J., O'Brien, J, Rodden, T., Rouncefield, M., Sommerville, I., (1995). "Presenting Ethnography in the Requirements Process", Proc. 1st Int. Conf. On Requirements Engineering, York, UK, March, pp. 27-34.

[6] Jackson, M. (2001). Problem Frames: Analyzing and Structuring Software Development Problems 1^{st} Edition, Addison-Wesley, pp.20.

[7] Landauer, Thomas K., (1996). "The Trouble with Computers: Usefulness, Usability, and Productivity, The MIT Press.

[8] Lewandowski, G., Gutschow, A., McCartney, R., Sanders, K., and Shinners-Kennedy, D. 2005. "What novice programmers don't know". In Proceedings of the 2005 international Workshop on Computing Education Research (Seattle, WA, USA, October 01 - 02, 2005). ICER '05. ACM Press, New York, NY, pp. 1-12

[9] Lilly, S.; (1999). "Use case pitfalls: top 10 problems from real projects using use cases Technology of Object-Oriented Languages and Systems". Proceedings TOOLS 30, 1-5 Aug. 1999 pp. 174 – 183.

[10] Pohl, K.; (1996). "PRO-ART: enabling requirements pre-traceability", Proceedings of the Second International Conference on Requirements Engineering, 1996, 15-18 April 1996 , pp. 76 – 84.

[11] Sommerville, I.; Rodden, T.; Sawyer, P.; Bentley, R.; Twidale, M.; (1993). "Integrating ethnography into the requirements engineering process", Proceedings of IEEE International Symposium on Requirements Engineering, 1993, 4-6 Jan. 1993 pp. 165 – 173.

[12] Sutcliffe, A., (2003). "Scenario-based requirements engineering", Proceedings 11th IEEE International Conference on Requirements Engineering, 2003. 8-12 Sept. 2003 pp. 320 - 329

[13] Tan, S. K. S., "The Elements of Expertise," Journal of Physical Education, Recreation & Dance, 68, 2, February, (1997), pp. 30-33.

[14] Viller, S. and Sommerville, I. (1999). "Social analysis in the requirements engineering process: from ethnography to method", Pceedings IEEE International Symposium on Requirements Engineering, 1999. 7-11 June 1999 pp. 6 – 13.

[15] von Knethen, A.; Paech, B.; Kiedaisch, F.; Houdek, F.; (2002). "Systematic requirements recycling through abstraction and traceability", Proceedings. IEEE Joint International Conference on Requirements Engineering, 2002. 9-13 Sept. 2002 pp. 273 – 281.

Table 1. A summary of results of conducting a comparative analysis of RE technology with proposed framework.

RE Tech.	Abstract	Contextualise	Conditional Knowledge
Ethnography	✗ Ethnography provides "real world" context which is often lacking in other technologies [5]. However, it is difficult to draw design principles and other abstract lessons from a technique that is concerned with the detail of a particular situation [14].	✓ The technology encourages requirements engineers to retain primary focus on those who live and work in the domain concerned thereby enable contextualisation [5, 11].	✗ Requirements engineers are able to assimilate knowledge but by the very nature of ethnography they are unable to do this "rapidly" [5].
Use case	✓ Use cases do help requirements engineers present their understanding in abstract way through UCD even though use case specifications can be lengthy [9].	✗ One of the limitation of use case notation has been found to be that it does not provide context [9]	✓ Use case diagrams (UCD) allow requirements engineers to rapidly alter their conceptual model of the problem domain. It also enables them to document quickly adapt the associated use specifications.
Scenarios	✗ Scenarios provide a detailed (non-abstract) view of a particular event sequence. They are only abstracted at later stages of development. Scenarios are related to models by a process of abstraction [12].	✓ Sutcliffe [12] states that scenarios are stories or examples of events as a grounded narrative taken from real world experience. These stories may include details of the system context (scenes).	✗ Scenarios provide a detailed description of particular events. While they can enable the requirements engineer to assimilate new knowledge and assign it relevant context, it is difficult to utilise this knowledge to alter existing conceptual models because of the narrowness of the description.
Traceability Matrix	✓ Traceability matrices enable abstraction by allowing requirements engineers to summarize the important points of a requirements work product i.e. relationships.	✓ Traceability matrices provide context in the sense that it allows requirements engineers to document dependencies [15, 4].	✓ Developing a traceability matrix will allow the requirements engineer to adapt to changes in requirements. Pohl [10] reports that traceable specifications are essential for consistent change integration and can lead to fewer errors during system development.