The Conundrum of Categorising Requirements: Managing Requirements for Learning on the Move

*+D. T. Haley, *B. Nuseibeh, *H. C. Sharp, +J. T. Taylor
**Department of Computing, +Institute of Educational Technology*
The Open University, Walton Hall, Milton Keynes, UK M K7 7AA
Email: D.T.Haley [at] open.ac.uk

Abstract

This paper reports on the experience of eliciting and managing requirements using the Volere shell and template (Robertson & Robertson, 2003) on a large European-based multinational project whose purpose is to create a system to support learning using mobile technology. We provide details about the project below, describe the Volere tools, and explain how and why we used a flexible categorisation scheme to manage the requirements. Finally, we discuss three lessons learned: (1) provide a flexible mechanism for organising requirements, (2) plan ahead for the RE process, and (3) do not forget the waiting room.

1. Introduction

A project with multiple stakeholders who are generating new requirements, commenting on existing requirements, and accessing requirements for design and implementation activities causes requirements engineers difficulties. These include ensuring a consistent capture mechanism for requirements, reconciling differences of opinion over what constitutes a requirement, and providing suitable access mechanisms to support the multitude of goals, backgrounds, and pre-existing biases of many partners. We describe here MOBIlearn, a large European research project for mobile learning that faced these problems.

This paper focuses on the surprisingly difficult problem of how to organise and categorise requirements. The conundrum was how to design a scheme for organising requirements that would satisfy the needs of most of the diverse stakeholders when it seemed impossible to do so. We found that while it was easy to "file" requirements using one of the 27 categories defined by the Volere template (Robertson & Robertson, 2003), it was difficult to retrieve them later. The effort expended eliciting and documenting requirements is wasted if the requirements cannot be located when needed. A search of the literature failed to reveal any directly relevant ideas. In fact, the literature supported our experience that classifying and categorising are non-trivial tasks.

The fact that different development team members have different needs complicates the process of establishing useful categories. For example, Anne, a researcher who wanted to compare requirements gathered by various techniques would need a category called *elicitation method*. Bob, another researcher, wanting to examine all of the requirements he elicited on a certain day would need a category called *date elicited*. A category called *hardware platform* would be useful for Carol, a developer charged with implementing all requirements for a laptop. Anne, Bob and Carol represent just three of many possible viewpoints.

We found a workable solution to the problem of determining appropriate categories by deciding *not* to decide. The rest of the paper explains this paradoxical choice.

Section 2 presents the MOBIlearn project. Section 3 explains how the project elicited requirements. Section 4 gives an overview of the Volere tools MOBIlearn used to document requirements and how they were adapted to suit the needs of the project. Section 5 discusses the difficulties in categorising and describes the database we designed to deal with the categorisation problem. Section 6 offers lessons learned. Although this project involves mobile learning, the lessons are applicable to any project that has more than a few requirements. Sections 7 concludes with a summary.

2. Background

This section discusses MOBIlearn, the three learning domains covered by the project, and the development method used by the project. Further information is available on the MOBIlearn website ("MOBIlearn,").

MOBIlearn is a large, multinational, European-funded research project involving more than 15 organisations from eight different countries, whose purpose is to provide a framework for improved learning using mobile technology. Pedagogical research about the effectiveness and usefulness of mobile learning is an important part of the work; as final deliverables, MOBIlearn will produce a set of requirements, pedagogical guidelines, best practices, and an architectural framework to support mobile learning. The system produced will be a prototype, or instantiation, of a state-of-the-art mobile learning environment validated by the research.

MOBIlearn has several complications, not least of which is the large number of team members who are geographically, linguistically, and professionally diverse. These international, multilingual partners from industry and academia contribute different perspectives and expertise, which contributes to varying preferences for looking at requirements.

Another complication of MOBIlearn is the tension inherent in the two types of project deliverables: reports that present research results and a working prototype. A desirable and necessary output of the research includes requirements that cannot be implemented during MOBIlearn due to unavailable technology or insufficient resources allocated to the project. Requirements engineers must resist the pressure from the developers to discard requirements that will not be implemented in *this* project but could serve as a roadmap for *future* projects.

The scope of the project adds additional challenges. MOBIlearn is concentrating on three types of mobile devices and three learning domains. The system delivered must work on laptops, smart phones, and PDAs. This multiplicity of hardware platforms provides issues of varying screen size, processing power, and existing software infrastructure.

The small budget available for the RE process was a contributing factor to difficulties we encountered. The two authors with RE experience were not funded at all by the MOBIlearn project and thus were constrained from spending a great deal of time. The other two authors have expertise in areas other than RE. Of these two funded, but inexperienced, authors, one managed several other areas besides RE and the author who was funded full-time for RE did not join until well into the project and after the developers had started to implement the MOBIlearn system.

Another ramification of the small budget available for RE was the inability to purchase an existing RE management tool. Much time was spent designing, implementing, and testing the requirements database created for MOBIlearn.

The three learning domains, or strands, provide the challenges of differing characteristics, needs, and types of learners. Each of the strands correlates to a type of learning. The museum strand typifies informal learning and concerns museum visitors, the most varied types of learners of our three strands. The MBA strand concerns formal learning by highly motivated, extremely busy professionals as well as first year beginning students. The health strand supports the need for periodic training and updating of skills of first aid workers.

3. Requirements elicitation in MOBIlearn

MOBIlearn started its design process by stating the overall purpose of the system – to support learners using mobile technology. It then conducted research (future technology workshops, questionnaires, observation, and interviews) to learn more about the tasks, types of learners, and interactions between learners and technology.

The museum strand used questionnaires to gather data from prospective visitors to the Uffizi Gallery in Florence. One interesting result from the questionnaire, although having nothing to do with learning, indicates a desire to have the ability to make and pay for reservations from a mobile device rather than the current system of waiting in line for hours. This requirement is an example of what unstructured questions on a questionnaire can produce: requirements that have nothing to do with the intended goal or purpose of the proposed system. We needed a way to categorise these unexpected requirements.

The MBA strand observed and interviewed students and educators to discover requirements. The requirements ranged from making and sharing annotations of PowerPoint slides to remote control of a classroom projector. Many of the requirements are implemented on widely available PDAs. We needed a way to separate requirements that implemented such familiar functionality from requirements that documented more innovative, MOBIlearn-specific features.

The members of the health strand conducted Future Technology Workshops (FTW) with first aiders to elicit requirements for the training and updating of their skills (Vavoula, Sharples & Rudman, 2002).. FTWs aim to explore the relationships between current and future technology for current and future activities. This elicitation technique produced many requirements, relating both to MOBIlearn's goal of supporting mobile learners and beyond its scope. For example, some first-aiders wanted a feature that would immobilise an injured person. Inventing such a technology is not part of the mandate of the project. Even so, we did not want to lose track of any requirement and needed to use a category for documenting even those requirements that we knew would not be implemented.

Maiden and Rugg (Maiden & Rugg, 1996) claim that requirements engineers should use a range of elicitation techniques. They suggest that scenario analysis, prototyping and RAD are the best techniques for new systems such as MOBIlearn. MOBIlearn used two of these techniques: scenario analysis and prototyping, in addition to questionnaires, observation, interviews, and FTWs. However, contrary to the advice in (Maiden & Rugg, 1996), we did not use a range of techniques for each strand because of time constraints. Our results might have been more complete if we had followed Maiden and Rugg's advice (Maiden & Rugg, 1996).

Possibly the reason that our questionnaire for the museum strand did not elicit particularly novel requirements is that, although collecting and analysing large amounts of data is easy, the structured format of a questionnaire inhibited creative thinking. Running a FTW might have been more productive.

Perhaps because we used a different technique for each strand, we obtained a large number of overlapping requirements e.g., users in each strand wanted to access a database. Some questions arose: should we keep three versions of a single requirement to document that it came from three sources? But since we are developing one product and not three, would it make more sense to keep one version with accompanying information giving the sources?

Another problem arose because each of the three techniques was used by a different set of requirements engineers, not all of whom used the Volere shell. They had differing views of what requirements should look like. One team delivered requirements that sounded like goals, e.g., "support the learner in everyday situations" but did not specify what the mobile system should do to fulfil this requirement.

These are just a few examples from our MOBIlearn experience that point out the need for a good classification system.

4. Requirements management in MOBIlearn

MOBIlearn adopted part of Robertson and Robertson's Volere process (Robertson & Robertson, 1999) for requirements elicitation and management: the Volere shell and template. Although the Robertsons incorporate the shell and template as part of their method, each can stand alone as an independent requirements documentation tool, which is how MOBIlearn used them.

The terms shell and template are confusing to some people. The Volere template is meant to be a guide for writing a complete requirements specification including all of the individual requirements. The Volere shell is also a template, but for a single requirement. Perhaps in response to this confusion, the Robertsons sometimes use the terms *atomic requirement template* and *shell* interchangeably. ("The Atlantic Systems Guild,").

Space does not allow for a complete description of these tools. See www.volere.co.uk for comprehensive information.

4.1. Uses of the Volere shell and template

The Volere shell provides a form for documenting requirements. It ensures consistency and compatibility in a clear and simple format. It affords traceability, both in where a requirement originates and where it appears in later documentation such as use cases. When used correctly and filled out completely, it encourages the originator of a requirement to study the detail of the requirement, justify the requirement, consider how it relates to other requirements, and how a tester can evaluate or test the requirement.

The Volere template is like a filing cabinet for storing requirements written on Volere shells. It comprises 27 categories of requirements, each of which is like a drawer in the filing cabinet. The purpose of the template is twofold: it is a template, or guide, for writing the final requirements documents and it serves as a checklist for the project (Robertson & Robertson, 2003).

4.2. Adapting the Volere shell

Because we found that the “out of the box” Volere shell did not completely satisfy our needs, we added two fields: status and title. The status field provides an easy search key. The title gives a short description that is useful for quick review of all the requirements. These simple additions helped enormously to locate particular requirements. Indeed, we could have added many more fields to the shell in an attempt to enable easier retrieval. Instead, we tried to adapt the Volere template for reasons discussed in the following sections.

The next improvement was more substantial. The Volere shells on 3x5 cards were not sufficient in themselves because the number of requirements quickly grew too large to manage without some kind of computerised requirements management system. Some of the many existing products that are available are: Doors (http://www.telelogic.com/), Calibre (http://www.calibresys.com/index.cfm), and the Rational product suite (http://www.rational.com/). Comparative reviews of these and other products can be found on ("The Atlantic Systems Guild,"), (Robinson, 2001), and (Young, 2002).

Because we did not have the budget to purchase an existing tool, we created an in-house database system. The MOBIlearn database was designed to offer online access with easy search and retrieval by our international team members. The initial version was based on the Volere template’s 27 types of requirements. We found that although it was easy to store a requirement and assign one of the 27 Volere categories, it was not easy to retrieve a particular requirement. The next section explains why.

4.3. Adapting the Volere template

The Volere template is similar to a filing cabinet with 27 very useful and relevant categories of requirements providing an organising principle for a requirements database. We “filed” each MOBIlearn requirement in one of the 27 “drawers”. However, the drawer for functional and data requirements became overstuffed.

We were troubled to discover that even 27 categories were not enough to provide useful search keys. We found that about 66% of our requirements were in one category – functional and data requirements. The first proposed change to the Volere template to make it more useful as a model for organising requirements in the database was to split functional and data requirements into separate sub-categories. This change was not sufficient because there were still about 64% of the requirements in the single category of functional requirements. We needed a very large “drawer” in our “filing cabinet” for functional requirements necessitating tedious one-by-one searching to locate the one we wanted. This discovery led to an attempt to subcategorise functional requirements as a means to improve the organisational structure of the requirements database. It was this attempt that revealed how difficult categorisation is.

5. Sub-categorising functional requirements

We tried several techniques to sub-categorise requirements. First, we used the “armchair” method (Furnas, Landauer, Gomez & Dumais, 1987), that is, we sat and thought about what made the most sense to us. After realising that many possible organising criteria exist, we reviewed the literature but identified no one who had solved the problem of deciding on just one method of classification. See Haley, et. al (2004) for a description of our attempt to use the card-sorting method. Finally, we decided not to decide, which resulted in our redesigned database. The following sections discuss these points in greater detail.

5.1. The difficulty of choosing categories

In order to provide easier access to the individual requirements in the database, we needed to break down the category of functional requirements to subcategories, which at first glance, seemed an easy task. On second and further glances, it became clear that it was not a trivial problem. In fact, the literature contains numerous accounts of the difficulty of categorisation. Haley et. al (2004) presents relevant research from four sources.

5.2. The problem of too many categories

Thinking back to our filing cabinet analogy, we see that we would need a filing cabinet for each categorisation criterion and copies of each requirement to file in each cabinet. Even if we decided to use such a cumbersome storage system, we would need to install a new filing cabinet and re-file each requirement every time a stakeholder requested a new organisation of the requirements.

These problems led to the conclusion that any database system used to manage requirements must provide a flexible view of the data. The next section describes the improved MOBIlearn database.

5.3. The MOBIlearn requirements database

The main innovation in the requirements database is a feature that allows flexible, non-static, and ad-hoc categorisations. Users with appropriate permission can add a new categorisation criterion at any time. Thus, the database can be modified easily to reflect new decisions, new information, and experience gained.

For example, at the beginning of the project we might have set up categories using the criteria of Strands (health, MBA, and museum) and of Work Packages (e.g., learning content, mobile media delivery, context awareness). These two criteria would reflect our belief that users would wish to query the database based on which requirements pertained to a particular strand or work package. After more experience, we might discover a need to find all requirements pertaining to a particular service. Later, managers may find that looking at requirements based on Work Package does not suit their needs because a particular Work Package may have members from different countries as well as different organisations. Managers could add a new criterion, location, to the database.

The disadvantage to this method is the need to add an additional piece of data to every requirement when a new categorisation scheme is adopted. Mitigating this disadvantage is that, with a well designed database, the process of adding data is straightforward even if time consuming.

This simple idea of providing the ability to modify the categorisation criteria of the requirements database enabled it to meet the needs of the many different project members throughout the life of the project.

6. Lessons learned

This section presents the lessons learned on the MOBIlearn project. Although MOBIlearn focused on support for learners on the move, the lessons are applicable to a wide range of projects.

Lesson 1: Provide a flexible mechanism for organising requirements

We found various problems that prevented us from using a static, predefined categorisation system for retrieving our requirements:

- People have varying points of view and want to examine the requirements from different aspects.
- These desires change over time and during different stages of the project.
- People don't always know what they want until they are deeply involved in the project.

The negative impact from these problems can be lessened by using a storage system for requirements that allows ad hoc updating of categorising criteria. As the number of requirements grows, it becomes necessary to give project members a more personalised view of the requirements.

Lesson 2: Plan ahead for the RE process

Our experience of managing requirements on the MOBIlearn project suggests that the RE process has requirements itself. If a requirements tool is not in place early in the project, practitioners will create their own, lightweight tools. The MOBIlearn database was not available until midway through the project resulting in duplication of effort. The first version of the database provided a view of the data organised according to the Volere template. One team member developed a spreadsheet to categorise requirements according to his needs, which were not satisfied by Volere. Another member created a sophisticated word-processing tool to organise information in yet a different format. These three tools, while not equivalent, contained a substantial overlap of information. This duplication of data results in duplicated effort in keeping the information up-to-date as well as increases the chance for errors, omissions, and conflicting data. Early selection or development of a requirements tool and commitment to using it can help keep projects on schedule by reducing wasteful effort.

Lesson 3: Do not forget the waiting room

Various tensions become apparent on a project like MOBIlearn with widely diverse stakeholders comprising both research-oriented academics and product-oriented industry representatives. In particular, we have noted an increasing tension between the researchers' desire to create an architectural framework and list of requirements for *future* implementations and the practitioners' desire to produce a functioning product *now*. The tension is natural because MOBIlearn's sponsor expects both types of results.

Practitioners want to limit the requirements to what they are able to deliver. The requirements engineers want to deliver a set of requirements for mobile learning regardless of whether they can be implemented with current technology and within the MOBIlearn time and budget constraints.

Requirements engineers can lessen the tension while preventing the loss of requirements that are not able to be implemented by using the category in the Volere template called *waiting room*. If circumstances change, either technological advances or budget constraints, any requirements stored in the waiting room are candidates for implementation.

7. Conclusions

We handled the conundrum of how to categorise the functional requirements after realising the impossibility of a system that would meet everyone’s needs. The MOBIlearn database provides a feature that allows ad hoc creation of new categorisation criteria. By deciding not to decide and allowing users to customise views of the database, we offered users a balance between flexibility and uniformity.

Our experience has reinforced the following points:

- Do not impose a static, predefined scheme for categorising requirements.
- Take time early in a project to address the requirements of the RE process.
- Use the idea of the waiting room to avoid losing track of interesting, but unable to be implemented, requirements.

8. Acknowledgements

We would like to acknowledge the European Union for financial support through the MOBIlearn project IST-2001-37440. Work partially supported by the European Community under the Innovation Society Technologies (IST) programme of the 6th Framework Programme for RTD - project ELeGI, contract IST-002205. This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing therein.

9. References

"The Atlantic Systems Guild," Http://www.systemsguild.com/. (23 Jan 2004).

G. Furnas, T. K. Landauer, L. Gomez, and S. T. Dumais, "The Vocabulary Problem in Human-System Communication," *Communications of the ACM*, vol. 30, no. 11, pp. 964-971, 1987.

D. T. Haley, B. Nuseibeh, H. C. Sharp, and J. T. Taylor, *Managing Requirements for Mobile Learning*, The Department of Computing, 2004/xx. The Open University, 2004.

N. Maiden and G. Rugg, "ACRE: A Framework for Acquisition of Requirements," *IEEE Software Engineering Journal*, May, pp. 183-192, 1996.

"MOBIlearn," Http://www.mobilearn.org/. (23 Jan 2003).

J. Robertson and S. Robertson, james | suzanne@systemsguild.com, "Volere Requirement Resources," Http://www.volere.co.uk. Atlantic Systems Guild, 2003. (15 September 2003).

S. Robertson and J. Robertson, *Mastering the Requirements Process*. Harlow, England: Addison-Wesley, 1999.

M. S. Robinson, "Requirements Management Tools: A Trade Study," Http://www.ess.stsci.edu/fset/projects/Sabbatical/, 28 March, 2001. (23 Jan 2004).

G. Vavoula, M. Sharples, and P. D. Rudman, "Developing the 'Future Technology Workshop' Method," in *Proceedings of Interaction Design and Children*, M. Bekker, P. Markopoulos, and M. Kersten-Tsikalkina, Eds. Eindhoven, The Netherlands, 28-29 August, 2002.

R. Young, "Requirements Tools Trade Study," Www.ralphyoung.net, 11 Jan, 2002. (23 Jan 2004).