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

D. T. Haley[*+] B. Nuseibeh[*] H. C. Sharp[*] J. Taylor[+]

[*] Department of Computing [+] Institute of Educational Technology
The Open University, Walton Hall, Milton Keynes, UK M K7 6AA
Email: {D.T.Haley, B.Nuseibeh, H.C.Sharp, J.Taylor}@open.ac.uk

Abstract

This paper reports on our experience of eliciting and managing requirements on a large European-based multinational project, whose purpose is to create a system to support learning using mobile technology. The project used an adapted database version of the Volere shell and template [9] . We provide details about the project, 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) use the 'waiting room'.

1. Introduction

A project with multiple stakeholders with different backgrounds, generating new requirements, commenting on existing requirements, and accessing requirements for design and implementation activities is a challenging prospect. Problems include ensuring a systematic approach to eliciting 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. In this paper, we describe MOBIlearn, a large European research project for mobile learning that faced, and largely resolved, these problems.

The paper focuses on the surprisingly difficult problem of how to organise and categorise requirements in such a fluid and dynamic environment. The conundrum was how to devise a scheme for organising requirements that would satisfy the conflicting needs of most of the diverse stakeholders. We found that while it was easy to "file" requirements using one of the 27 categories defined by the Volere template [9], different stakeholders did not find this categorisation helpful when accessing the requirements to support their work. The effort expended eliciting and documenting requirements is wasted if the requirements cannot be located when needed. A search of the literature and discussion with other Volere users failed to reveal any directly relevant experiences. In fact, the literature supported our experience that classifying and categorising are non-trivial tasks.

The fact that different development team members have genuinely 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. The concerns of Anne, Bob and Carol represent just three of many possible perspectives on the requirements.

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 that 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. Section 7 concludes with a summary.

2. Background

MOBIlearn [5] is a large, multinational, European-funded research project involving more than 15 organisations (a mixture of academic and industrial enterprises) 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 of a state-of-the-art mobile learning environment validated by the research.

MOBIlearn faces a number of challenges, 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 challenge 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. Analysts 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 introduces 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 personal digital assistants (PDAs).

The three learning domains, or strands, were articulated through scenarios, and 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.

It is in the nature of European funding models that permanent members of staff in educational establishments are not funded by the project. Thus, a researcher was contracted to work on requirements several months after the project had started, to bring together, and add weight to, efforts that had been made to engage project members in requirements gathering. One of the issues for the project was that requirements engineering (RE) was not regarded as a primary activity by many partners, and the domain in which the project is located (mobile learning environments) is sufficiently new that there is little prior experience to build on. As a research project, there was also no budget available to purchase an existing requirements management tool, so we were looking for a lightweight, but effective, way of using existing software to deal with the capture, recording and retrieval of requirements.

3. Requirements elicitation in MOBIlearn

MOBIlearn began by stating the overall purpose of the project – 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. Requirements gathering for each strand was conducted by a different project partner.

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 be able to make and pay for reservations from a mobile device rather than the current system of waiting in line for hours. We needed a way to categorise these sorts of 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 project 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 [12] . 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 [4] recommend REs use a range of elicitation techniques. They claim scenario analysis, prototyping and RAD are the best techniques for new systems. MOBIlearn used the first two of these techniques in addition to questionnaires, observation, interviews, and FTWs. However, due to variations in the user populations, in the environments in which the research took place, and time constraints, it was not possible to run all techniques for each scenario strand.

Perhaps because we had to use different techniques for each scenario strand, we obtained a large number of overlapping requirements; e.g., users in each strand wanted to access a database. Should we keep three versions of a single requirement to document that it came from three sources? Or, alternatively, since we are developing one product and not three, would it make more sense to keep one version with accompanying information giving the sources?

Furthermore, each of the techniques was deployed by different researchers from various backgrounds, none of whom had ever used Volere. They had differing views of what requirements should look like. So, for example, requirements were sometimes expressed as goals; e.g., "support the learner in everyday situations" but did not specify what the mobile system should do to achieve this goal.

4. Requirements management in MOBIlearn

From an RE perspective, our task was to support these researchers, who were inexperienced in RE and who had limited time available. This was another reason for not using a commercial requirements management tool due to layers of complexity that would not be appreciated by our partners.

We felt that a reasonable approach to trial was to adopt part of Robertson and Robertson's Volere process [10] for requirements elicitation and management: the Volere shell and template. Although the Robertsons incorporate the shell and template as part of their process, each can stand alone as an independent requirements documentation tool, which is how MOBIlearn used them. The Volere template was meant as a guide for writing requirements specifications including all of the individual requirements [9].

4.1. Uses of the Volere shell and template

The Volere shell provides a form for recording requirements. It is intended to ensure consistency of presentation in a simple format. In principle, it affords traceability, both in respect of where a requirement originates and where it appears in later documentation such as use cases.

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. Ideally, 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 assess how a tester can evaluate or test the requirement. But the time commitment required to completing the template is not negligible, and our researchers frequently found it difficult and time consuming. So, we needed to adapt the shell to suit the purposes of the project, and to encourage people to use it.

4.2. Adapting the Volere shell

Because we found that the "out of the box" Volere shell did not completely satisfy our needs, we decided to create a database template from the shell, but added two extra fields: status and title. The status field allows researchers to look at requirements in various states of completion; e.g., all requirements in the process of being refined. 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. The database, with this front end, enabled our partners across Europe to inspect the requirements so far, and to create or modify entries as and when they were able to work on them. In principle, the idea was that this would prevent a bottleneck, as all partners who asked for access could have it, and it supported the idea of distributing responsibility for requirements across all members of the project. Furthermore, it made requirements contributed so far a common resource that all could inspect, so discussions could take place around specific requirements and their development. We hoped this would make the task of refining initial attempts easier.

The database was prototyped quite quickly and offered to the researchers in the field who began using it and feeding back comments on usability and recommendations for changes to the development team. This process is on-going, and, in tandem with the development of the scenarios, is progressing well.

The initial version of the database was based on the Volere template's 27 types of requirements. We found that, although it was straightforward to store a requirement and assign one of the 27 Volere categories, the stakeholders did not find this categorisation helpful when retrieving a particular requirement. The next section explains why.

4.3. Adapting the Volere template

We filed each MOBIlearn requirement in one of the 27 "drawers". However, we found that about 66% of our requirements were in one category – functional and data requirements – and the 27 categories were not appropriate to provide useful search keys. So, the first proposed change to the Volere template to make it more useful for organising requirements 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 categorise functional requirements further as a means to improve the organisational structure of the requirements database.

5. Categorising functional requirements

We tried several techniques to categorise requirements. We began by using an "armchair" method [3], that is, we sat and thought about what made the most sense to us. However, we soon realised that many possible organising criteria exist, and found no related work to address this issue. So, we decided not to decide, which resulted in our redesigned database.

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 sub-categories, which at first glance, seemed an easy task. On reflection it became clear that it was not a trivial problem. In fact, the literature contains numerous accounts of the difficulty of categorisation.

Dumais and Landauer [2] describe two kinds of categorisation problems for menu-based information retrieval systems. Their research applies to the MOBIlearn database. First, they identify the difficulty of selecting a category name that can be agreed upon and understood by most users. Second, the categories that most people use are blurry and not mutually exclusive. Our experience supports their findings.

Bowker and Star [1] observe that designers of classification systems from different disciplines will have different needs resulting in systems where just one point of view is legitimized and others are ignored [1]. They observe it is impossible for designers to know in advance what information will be relevant in the future [1]. They claim that categorising for information systems is particularly difficult because designers must consider both "the hard technical problems of storage and retrieval with the hard interactional problems of querying and organizing" [1]. Again, our experience supports their findings.

Rugg and McGeorge implicitly acknowledge the difficulty of categorising, and describe card-sorts as a knowledge elicitation technique [11] . They describe a study in which subjects came up with five different categorisation criteria when shown pictures of a reasonably familiar object. This supports our difficulty in devising a single criterion-based system to categorise functional requirements.

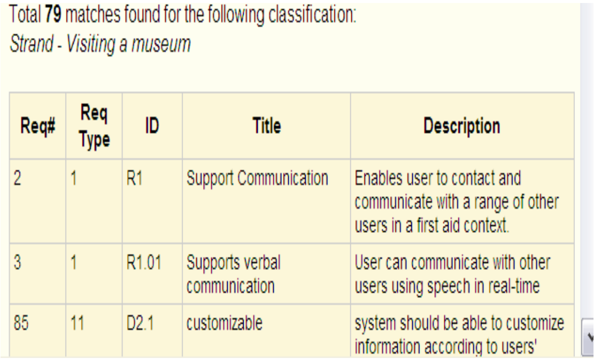
Prieto-Diaz [8] has been investigating the problem of categorising for about 20 years [6] (as cited in [7]) using faceted lists in his classification scheme. His work focuses on software component reuse but many of his findings apply to RE as well. For example, he observes that any classification scheme must allow for continuing growth in the items being classified [7] , a condition that certainly applies in RE.

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 database is a feature that allows flexible and ad-hoc categorisations. Users with appropriate permission can add a new categorisation criterion at any time to reflect new perspectives, decisions, information, and experience. For example, at the beginning of the process, we set up categories using the criteria of project Strands and of project Work Packages. Figure 1 shows a partial report listing three of the museum strand requirements. After more experience, we discovered a need to find all requirements pertaining to a particular service. Later, some partners found that looking at requirements based on Work Packages did not suit their needs because some Work Packages had members from different countries and different organisations. Partners could then add a new criterion, location, to the database.



Total **79** matches found for the following classification:
Strand - Visiting a museum

Req#	Req Type	ID	Title	Description
2	1	R1	Support Communication	Enables user to contact and communicate with a range of other users in a first aid context.
3	1	R1.01	Supports verbal communication	User can communicate with other users using speech in real-time
85	11	D2.1	customizable	system should be able to customize information according to users'

Figure 1: Museum strand requirements

The disadvantage of this technique is the need to introduce 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, if time consuming. Figure 2 shows a requirement being categorised in the database. The possible categories are shown on the left. The researcher has selected “Health” from the possible strands; WP2 and WP4 from the possible work packages; GE from the possible services; text, audio, and graphics from the possible media; and PDA from the possible hardware platforms. The process of choosing categories is easy because the user does not need to remember any of these categories and needs to make a minimum of clicks – only 13 to make all the choices just described.

This idea of providing the ability to modify the categorisation criteria of the requirements enabled it to meet the needs of the many different project members throughout the life of the project. It also illustrated how such a flexible resource can support the conversation between requirements and implementation, each influencing the other as time progresses.

6. Lessons learned

MOBIlearn focused on support for learners on the move, although we believe that 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 concerns and want to examine the requirements from different perspectives.
- These concerns 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 adverse impact of these problems can be lessened by using a requirements management system that allows ad hoc updating of categorisation criteria. As the number of requirements grows, it becomes necessary to give project members a more personalised view of the requirements. Ongoing development of the database includes a versioning facility to track and review the development of a particular requirement.

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 before requirements are ready to be documented, practitioners will create their own, lightweight tools. 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 resulted in duplicated effort in keeping the information up-to-date as well as increasing the chance of errors, omissions, and conflicting data. Early selection or development of a requirements tool and commitment to using it can help keep projects on schedule.

Lesson 3: *Use the ‘waiting room’*

Various tensions become apparent with widely diverse stakeholders comprising both research-oriented academics and product-oriented practitioners. In particular, we 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. Analysts want to deliver a set of requirements for mobile learning regardless of whether or not they can be implemented with current technology and within the project time and budget constraints.

To prevent the loss of requirements that cannot be implemented, Volere uses a template category called the *waiting room*. If circumstances change, either technological advances or budget constraints, any requirements stored in the waiting room are candidates for implementation.

7. Conclusions

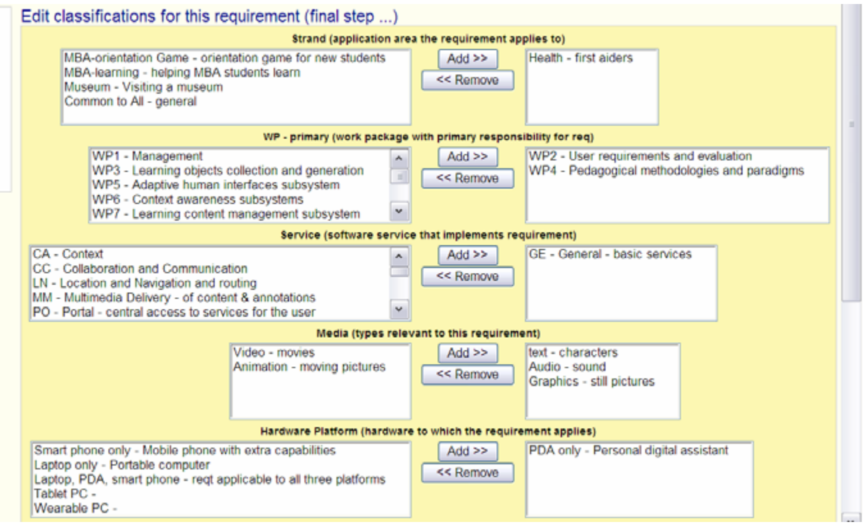
We faced the conundrum of how to categorise functional requirements after realising that we could not develop a static system that satisfied everyone. The MOBIlearn database provided ways to facilitate ad hoc creation of new categorisation criteria. By deciding not to decide and allowing users to customise views of the database, we offered users a balance between flexibility and uniformity. Work continues, but our experience so far has reinforced the following lessons:

- 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 requirements that are interesting but are not scheduled to be implemented.

Acknowledgements. We acknowledge the EU for financial support through the MOBIlearn project (IST-2001-37440) and the ELeGI project (IST-002205). The views expressed in this paper are those of the authors and may not represent the views of the EU.

8. References

1. G. C. Bowker and S. L. Star, *Sorting Things Out: Classification and Its Consequences*. The MIT Press, 1999.
2. S. T. Dumais and T. K. Landauer, "Using Examples to Describe Categories," in *Proceedings of CHI'83*. ACM, 1983, pp. 112-115.
3. G. Furnas, T. K. Landauer, L. Gomez, and S. T. Dumais, "The Vocabulary Problem in Human-System Communication," *Communications of the ACM*, 30(11), 964-971, 1987.
4. N. Maiden and G. Rugg, "ACRE: A Framework for Acquisition of Requirements," *IEEE Software Engineering Journal*, 183-192, May 1996.
5. "MOBIlearn," http://www.mobilearn.org/. (23 Jan 2003).
6. R. Prieto-Diaz, "A Software Classification Scheme," Ph. D. Diss. Department of Information and Computer Science: University of California, Irvine, 1985.
7. R. Prieto-Diaz, "Implementing Faceted Classification for Software Reuse," in *12th International Conference on Software Engineering*. IEEE CS Press, 26-30 March, 1990.
8. R. Prieto-Diaz, "A Faceted Approach to Building Ontologies," in *IEEE International Conference on Information Reuse and Integration*. IEEE CS Press, 2003, pp. 458-465.
9. J. Robertson and S. Robertson, "Volere Requirement Resources," http://www.volere.co.uk. Atlantic Systems Guild, 2003. (15 September 2003).
10. S. Robertson and J. Robertson, *Mastering the Requirements Process*. Harlow, England: Addison-Wesley, 1999.
11. G. Rugg and P. McGeorge, "The Sorting Techniques: A Tutorial Paper on Card Sorts, Picture Sorts and Item Sorts," *Expert Systems*, 14(2) 80-93, May 1997.
12. 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.



Edit classifications for this requirement (final step ...)

Strand (application area the requirement applies to)

MBA-orientation Game - orientation game for new students
MBA-learning - helping MBA students learn
Museum - Visiting a museum
Common to All - general

Health - first aiders

WP - primary (work package with primary responsibility for req)

WP1 - Management
WP3 - Learning objects collection and generation
WP5 - Adaptive human interfaces subsystem
WP6 - Context awareness subsystems
WP7 - Learning content management subsystem

WP2 - User requirements and evaluation
WP4 - Pedagogical methodologies and paradigms

Service (software service that implements requirement)

CA - Context
CC - Collaboration and Communication
LN - Location and Navigation and routing
MM - Multimedia Delivery - of content & annotations
PO - Portal - central access to services for the user

GE - General - basic services

Media (types relevant to this requirement)

Video - movies
Animation - moving pictures

text - characters
Audio - sound
Graphics - still pictures

Hardware Platform (hardware to which the requirement applies)

Smart phone only - Mobile phone with extra capabilities
Laptop only - Portable computer
Laptop, PDA, smart phone - reqt applicable to all three platforms
Tablet PC -
Wearable PC -

PDA only - Personal digital assistant

Figure 2: Selecting Categories