Requirements Engineering in the Health Care Domain

Luiz Marcio Cysneiros [1]
Department of Mathematics and Statistics
Information Technology Program
York University
cysneiro@mathstat.yorku.ca

Abstract

There are many different approaches to elicit requirements each having its strengths and weaknesses. Hence, some approaches may be ore suitable to one domain then another. Moreover, some domains may require these approaches to be carefully applied or even adapted to work efficiently. Health care domain is one of these domains. It is a complex domain with many subtleties, such as political and legal issues that have to be taken into account. This work brings some of the lessons learned in more than six years working in several hospitals and laboratories. Particularly, this paper presents some elicitation techniques that had to be adapted in order to comply with the constraints imposed by several peculiarities intrinsic to this domain. It also points out some special considerations that must be taken into account regardless the method one chooses to elicit requirements.

1. Introduction

As we face more complex systems, more challenging is the requirements engineering job. To deal with this challenge, many work have been done trying to improve requirements elicitation process and techniques. Usually, these processes and techniques aim to work in a broader spectrum and thus they are not concerned about a particular domain.

Although understandable, it may not be enough when one deals with particularly complex and unknown domains such as the health care domain. Health care organizations have been demanding a lot of effort to be put on developing several types of software. Most of these software are critical because they involve peoples life in a large and continued scale and therefore are very sensitive to mistakes, even more if these mistakes come from requirements elicitation.

In the health care domain we deal with many constraints to elicit requirements, constraints that range from the availability of stakeholders to the restrictions on the use of some elicitation techniques due to legal issues. Some of these constraints can be quite challenging to overcome, and if not dealt with, may lead to a wrong set of requirements.

Recently, the author has been noticing a growing number of Ph.D. theses, mostly from Europe, that in some way use facilities in the health care domain. By reading these thesis and talking to other people involved in developing software for the health care domain, it was possible to realize that many of the pitfalls that the author have encountered before were faced during these works.

This work aims to point out some of these pitfalls and possible ways of dealing with them. It is based on the experience gained during more than six years working in requirements elicitation as part of real life projects where we used many different approaches, ranging from structured analysis [1] to goal-oriented requirements [2][9], and techniques ranging from semi-structured interviews to ethnography. The paper presents many lessons regarding some concerns one might have regardless the methodology being used for eliciting requirements as well as adaptations to some elicitation techniques used [3]. During these six years the author has participated in the software development process within three different hospitals in Brazil, three different hospitals in Canada and four different clinical analysis laboratories in Brazil. Hence, it is fair to say that the experience gathered is quite comprehensive and might allow this work to express behaviors and needs that might be found in several organizations within the health care domain.

It is true that some of the aspects that will be shown in this paper can be at least partially true for other complex domain, but they are still a matter of concern to those involved with the health care domain and therefore, bringing these aspects to light can contribute to avoid some common pitfalls.

It is important to mention that this work is not supposed to tackle problems related to eliciting requirements for areas involved with bioinformatics such as molecular biology, gene sequencing and protein

[1] This work was partially granted by CNPq

expression. This is a completely different area with particularities that hold only for this kind of software.

Section 2 will focus in some of the special attention one may have regarding the requirements elicitation process, while section 3 will tackle the different elicitation techniques that were used and adapted. Section 4 will conclude the paper.

2. Special Considerations to the Requirements Elicitation Process

This paper does not focus on any particular approach to requirements elicitation. It is understood that all of them have their strengths and their weakness. Hence, we will try to bring to light some concerns that might deserve special attention regardless the process being used.

2.1. Understanding the Vocabulary Used in the Domain

The first problem one may face in the health care domain can be very challenging: to understand what the stakeholders are saying. Although the need for using some kind of vocabulary control has been acknowledge as an important step in requirements elicitation for any domain [4][5], here it plays a crucial role because the vocabulary used tends to be unfamiliar to most of the requirements engineers. The main idea behind the use of a glossary is that before modeling any domain, one has to understand the vocabulary used in this domain. Therefore, capturing it in an organized form not only come at a low cost but also serves to facilitate the reuse of the domain's knowledge. It has been particularly useful when new engineers join the development team during the software development process. Most of the organizations that were used as basis for this papers expected new software developers joining the development team to take at least three months before they are really capable of producing any reliable specification. After the vocabulary control was introduced this expectative dropped to one month. Not only they were able to learn from the existing vocabulary, as they did not have to go to stakeholder every time they faced an unknown term.

When eliciting requirements in the health care domain, one will always be facing a lot of unknown and sometimes weird (at least for us) terms. Even worse, in this domain one single term frequently has more than one meaning depending on the stakeholder that is talking to you. The meanings are not only different but also sometimes conflicting. Take for example a clinical analysis laboratory. One stakeholder had the understanding that *"to electronically sign a patient report"* means that once all the results are available, the physician would review the patient report authorizing the system to print this report with the physician's signature. Another stakeholder understood the same term meaning that the system must able to reason whether all the test results are within a safe range of values allowing the system to automatically assign this result to the patient's report or not. If all the results are within this safe range, the system must print the report with the physician's signature. These are two different interpretations to the same term that could lead to quite different software requirements.

To tackle this problem is suggested the use of some kind of vocabulary control. In particular, the Language Extended Lexicon (LEL) [5] was used in almost all the cases studies that were carried out with good results. The objective of the LEL is to register the vocabulary of a given UofD[2]. It is based upon the following simple idea: understand the problem's language without worrying about deeply understanding the problem. The main objective of the LEL is to register signs (words or phrases) peculiar to a specific field of application.

The LEL is based on a code system composed of symbols where each symbol is an entry expressed in terms of notions and behavioural responses. The notions must try to elicit the meaning of the symbol and its fundamental relations with other entries. The behavioural response must specify the connotation of the symbol in the UofD. Each symbol may also be represented by one or more aliases and will be classified as a subject a verb or an object.

The construction of the LEL must be oriented by the minimum vocabulary and the circularity principles. The circularity principle prescribes the maximization of the usage of LEL symbols when describing LEL entries, while the minimal vocabulary principle prescribes the minimization of the usage of symbols exterior to the LEL when describing LEL entries. Because of the circularity principle, the LEL has a hypertext form.

To build an initial version of the LEL we start performing an open-ended interview with some or all of the stakeholders (depending on either their availability and the political convenience of doing that). During this meeting we will try to get a first idea of the domain and to capture some initial LEL symbols. We will also try to identify possible documents that we can use to further elicit other symbols. The use of other types of interviews and protocol analysis [3] can be particularly useful here. Have in mind that every time a technique is mentioned it will be mentioned assuming the adaptations that will be presented in Section 3.

Figure 1 shows an example of a LEL symbol. In this figure, it is shown a term extracted from one case study

[2] *"Universe of Discourse is the general context where the software should be developed and operated. The UofD includes all the sources of information and all known people related to the software. These people are also known as the actors in this UofD."*

conducted at three major hospitals in Toronto area. It can be seen in this figure that this symbol has two other synonyms. Initially we were thinking that attending physician and admitting physician were two different symbols expressing two different roles in the domain. When we validated the lexicon with the stakeholders they pointed out that actually, the two symbols had the same meaning and therefore were synonymies. The other alias that can be seen in this symbol is just to address the problem of letting the tool handle plurals. All the underlined words are other symbols of the LEL establishing a hypertext format to the LEL that makes it easy to navigate through symbols.

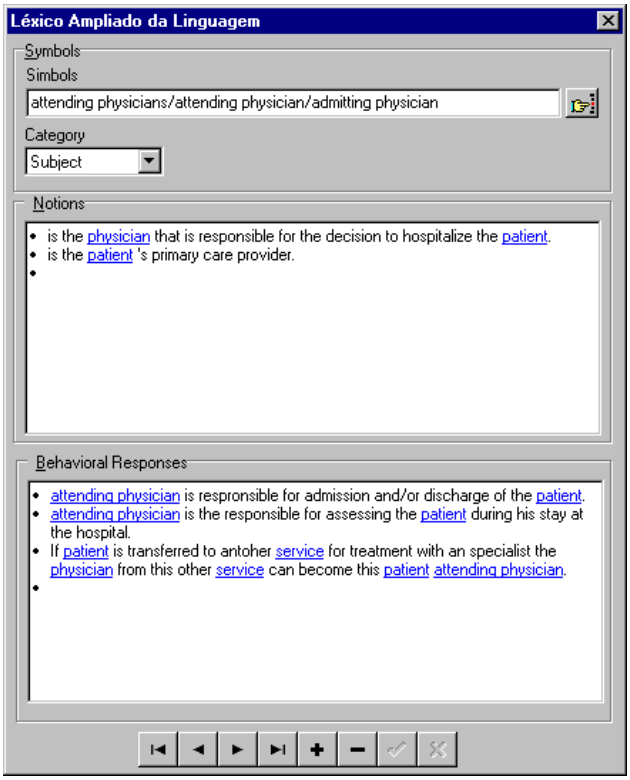

Figure 1 – Example of the LEL

2.2. Look For Different Ways of Doing Things

Looking for different alternatives for the current process is an important step of the requirements elicitation process regardless the application domain. Nevertheless, it is even more important in the health care domain.

Only recently hospitals and laboratories have started their efforts in seriously using information systems in their business. Thus, although unusual, it is possible to find some facilities that have no software at all. Actually, it is more likely to find softwares that were developed targeting only the automation of an existing process. However, by only automating the existing process may frequently not solve the problem at hand or at least provide the stakeholder with less then the ideal process. In fact, due to the lack of efficient use of information technology, the health care domain is very suitable to apply Davenport's ideas. Davenport [6] claims that information technology should support the reengineering of a business so it can achieve the most efficient way of conducting a business. Although it might be true for virtually any domain, in health care domain it is particularly important, since many improvements can be made due to the use of information technology. More precisely, it is possible not only to increase the efficiency of a hospital or laboratory in financial aspects but also, and probably more important, the use of information technology can lead to a better and safer process.

For example, the Guardian Angel Project [7] proposes to apply information technology to support a shift from provider-centered care to patient-centered care. In this new approach patients would be the owner of their medical records, being able to carry these records to any hospital or physicians that he/she wants. Moreover, it proposes the use of PDAs to run intelligent software that could help on assessing the patient needs regarding some specific diseases in a long term basis taking into account the plan established by the physician but also individual preferences.

Another example can be drawn from a case study conducted at three major hospitals in Toronto area. We were told to study how the discharging process of these hospitals could be improved through the use of a software that would replace the existing one. After understanding and modeling the process and evaluating possible alternatives, we concluded that aside from a new software, major changes to the process itself would have to be done to solve the problems they wanted to solve by simply replacing the software.

Thus, it is important that the requirement engineer always search for new ways for achieving the same goals that are achieved today. It is important to understand the reasons that underlie the process, why things are done that way and what are the positive and negative aspects of doing things the way they are done today. Understanding these aspects can drive the requirements engineer to think in alternative and more efficient ways of achieving the goals.

Different alternatives to the process might as well address different viewpoints. Surprisingly, up to now most of the software built to work in the health care domain do not take into account the most important consumer of health care: the patient [8]. Software developers and health care providers tend to ignore that the final consumer is the patient and hence patient's concerns should be taken into account. For example, a clinical analysis laboratory may establish that the system would ask for drawing one blood sample for each

different sector within the lab where the tests will be processed. Thus, if a physician asks for eight different tests it is not unusual to use four different samples. However, many patients may not feel comfortable with that. The software engineer may try to evaluate if some kind of tradeoff can be done by eventually introducing a new step in the process for the cases where too many samples have to be drawn. During this step some blood could be taken from on sample and placed into a new one manually (or eventually using an automated process).

There are some approaches that can support these kind of reasoning such as the ***i**** Framework [2] and KAOS [9]. These two approaches can also handle another very important aspect in Health Care domain: non-functional requirements elicitation.

2.3. Eliciting Non-Functional Requirements

Many work have been showing the importance of non-functional requirements elicitation [9, 10, 11]. In health care domain they are not only important but also numerous. In fact, one of the most common sentences one can hear from the stakeholder ("We are dealing with the patients life") brings the safety non-functional requirement to the light. It is safe to say that in this domain, the requirements engineer must always be looking primarily for safety constraints that might apply to the process and hence to the software.

To address safety, the requirements engineering must also be asking himself and the stakeholders for each functionality of the system what safety concerns might one have regarding this functionality. The requirements engineer might try to reason about possible consequences that might arise if the system fails to provide this functionality.

Standards enforced by regulatory agencies or legal requirements may constantly be searched to prevent the system from breaking any rules. For example, many countries require that medical records do not show any changes, meaning that nothing can be erased or altered. To comply with these legal requirement it is necessary to have a system that store all patient's data in a historical basis so one can retrieve every step regarding a patient's treatment. It may also call for a non-trivial log of activities to be provided by the system in order to show who has inputted which data to the patient's record.

Security is also very important. The requirements engineer must always try to find out if there is any special concern regarding access control. There should be some special concerns like for example:

- Only employees that work in the Hematology section can input results for tests processed within Hematology.
- Only supervisors can input results for some tests when their results are out of a certain range.
- Only the attending physician can authorize the discharging of a patient.

Another important non-functional requirement is Privacy. Patients expect to have their privacy assured when they go to hospitals and laboratories. Omitting names from everywhere it can be omitted is frequently a concern. For example, when the information system is used to upload information to any equipment, patient's name should always be omitted. The same way, any written document that would be used o guide the work within the organizations, e.g. working lists, medications distribution, etc, might not contain names. Imagine the situation where a famous actress has to be tested for HIV virus for some reason. If her name appears in an analyzer or in a patient's report or in a work list together with the HIV test name, it is very likely that two days later the whole press will know about the test and at that time they will probably "infer" that the result was positive.

Reliability is also a major issue for a software in this domain. People who work in this domain usually are very sensitive about mistakes and do not understand very well that a software can somehow failure. Therefore, as in some other domains where reliability is a critical issue, an intensive test program should be established, preferably duplicating the current process within the proposed system, i.e. getting the new software to work in parallel with the process as it is carried out at the present moment comparing the results from both approaches.

Availability of information can be quite important in this domain since a physician may not have the time to wait for the system to reboot. It would be equally inconvenient if patient's data were not available when an analyzer processes a sample since this analyzer would simply skip the patient.

Usability is of course, as always, a must. Many users in this domain are not experienced with computers and therefore usability has to be a major concern. Haris [12] points out that although the UK has adopted the use of computer system as a national policy for professionals involved in assessing elderly people, only 1% of these professionals are currently using computer systems. It is an example of how difficult using a computer can be in this domain. It is acknowledge here that United States and Japan may not experience this constraint at least not so strongly as others, since health care facilities in these countries have been dealing with information technology for a while.

Therefore, one of the great challenges in the health care domain relies in designing a system that can be comfortable for both nurses and physicians to use. They have been using different forms of documents for centuries hence, giving up of this "tradition" is not an easy job. Moreover, they can easily carry these documents with them wherever they go. Furthermore, reading these documents or making any annotations to

them requires almost no effort. Therefore, the system must be designed targeting these characteristics. Better yet if usability aspects are designed together with the end users. Doing so might contribute not only for a good design but also to give nurses and physicians some sort of partnership in the system, which frequently contributes to diminish resistances to use the system in the future.

Due to the uniqueness of the environment and the work routine, sometimes usability aspects might fall into a complete different set of solutions from those that would be usually adopted. Just as an example, we have once considered providing one of the tasks that were performed within a system for clinical analysis laboratory with the capability for scanning bar code labels, to prevent one from inputting a 12 digit number. Of course, besides the usability aspect there was also the reliability aspect, once typing a 12 digits number may lead to mistakes. When we presented this idea to the stakeholders some of them pointed out that this solution would be impossible for them, because there was no space available in the workbench to place a bar code reader. A pen was not an option not only because it frequently fails to scan the label but also because the nature of the chemicals used in the bench could easily damage the pen. Thus, do not be surprised to find users that feel more comfortable using character-oriented interface than graphical interface. The use of the mouse can also be a challenge in some places

That brings one more important point. Non-functional requirements are frequently conflicting among them [10] therefore, since in the health care domain one may expect to find a large number of non-functional requirements, it is very important to search for conflicts. Some heuristics for doing that can be found in [11].

The aspects mentioned above can be seen as guidelines to be followed regardless the methodology one might use for requirements elicitation. Many elicitation techniques might be applied to gather information and some of them will suite better than other to the health care domain.

3. Elicitation Techniques

Many elicitation techniques have been proposed up to now. Each of them may more suitable for one particular situation than another and therefore, part of the requirements engineering challenge is to choose the right one for each moment. During these years working in the health care domain it was possible to observe that many techniques cannot be applied exactly the way it is originally described. Rather, some changes must be done to these techniques so one can use them efficiently. It is shown below some of the techniques that were more frequently used with good results together with some suggested modifications where applicable.

3.1. Document Reading

Using existing documents, job descriptions, task descriptions and quality assurance manuals among other type of documents can be of enormous help in this domain. Of course other domains can benefit from document reading also, but due to the use of a vocabulary that is totally unfamiliar to requirements engineering (at least most of them), reading documents can be the easiest way of starting to get contact with this new vocabulary.

3.2. Questionnaire

The use of questionnaires can be quite difficult in this domain. Here, a questionnaire is understood as a set of questions that are sent formally or informally to different stakeholders to be answered by them in a convenient time. Three different laboratories and two different hospitals were tested on the use of questionnaire as an efficient way of gathering requirements. Less then 10% of the distributed questionnaires were handled back. Within those ones, more than 80% were only partially filled up. These numbers are a general compilation of all the five organizations involved but they do not change expressively from one to another. Although this is a known weakness of questionnaires, it seems to be particularly true in this domain. No specific reason could be found to explain that, but it may be at least partially related to the fact that quite frequently nurses and physicians work in 12 or 24 hour shifts and frequently, the initial hours of theses shifts are work intensive, trying to catch all the work that was previously done by the other team.

However, when applied to the upper management at all five institutions the results were much better, ranking 84% of questionnaires returned, most of them satisfactorily filled up. Hence, the use of questionnaires may be worth in an explorative phase with the upper management.

3.3. Interviews

There are many types off interviews that can be applied in different situations [13]. Structured interviews for example, require the requirements engineer to formulate a previous list of questions to be followed during the interviews. Since it requires some knowledge of the domain, this type of interviews would be ore likely to be used in later phases of requirements elicitation. Open-ended interviews are more informal, giving more space to the stakeholders to answer the way they wish. In Conversation analysis [3] the order of interaction is negotiated in real-time.

Although structured interviews were used, the best result came from the use an intermediary approach between open-ended interviews and conversation analysis. The best approach was to prepare a list of topics that we wanted to address during the interview, but without imposing any limitations to the process, allowing free (although organized) interaction among the participants, with the order of interaction being negotiated in real time. Professionals in the health care domain can be quite suspicious of interviewer's integrity, not to mention the frequent fear for being sued because of something he/she might say during an interview. Using an approach that is closer to a informal conversation helps on gaining confidence and setting spirits free of many of their fears. The biggest challenge with this approach is to keep the interview in the right track, not allowing the "conversation" to get into undesirable detours that would lead you nowhere. Judging what is a desirable detour or not is far from clear and might demand some experience in the domain.

3.4. JAD Sections

JAD Sections [14] were used much more to solve conflicts than to gather knowledge about the domain. During these sections some or all the stakeholders are grouped trying to achieve some cooperation among them and the requirements engineering to jointly produce system's requirements. During our work we found out to be particularly true what is stated by [3]: "because participants may have different status within the organization, there is a danger that some will not feel free to say what they really think, specially if it is unpopular". As the author have used JAD sections in other domains with better results, it seems reasonable to assume that health care domain may not be very suitable to use JAD sections. One of the reasons to that might be the very competitive environment that is quite frequently found in this domain, together with the ethical problems that underlies almost all conversations. In spite of these problems, we could sometimes get good results using JAD sections to deal with conflicting requirements that involved more than two stakeholders.

3.5. Protocol Analysis

Protocol Analysis [15] consists on asking the stakeholders to talk aloud about what they are doing while they are performing some task. In its strict sense can be very inconvenient in the health care domain for many reasons. First, Hospitals are expected to be a quite place, and a requirement engineer following a physician while he talks about what he is doing can be quite inconvenient. Second, physicians often have no time to explain what they are doing and why they are doing while performing his duties. Imagine a physician assessing a cardio-respiratory failure and having to talk to you what he is doing. It would be funny if not tragic. Third, being highly concentrate in their jobs is a major requirement for both hospitals and laboratories employees. Having to talk loud would definitely brake this concentration and hence would interfere with the process.

To cope with these problems we have introduced a third part to the protocol analysis technique. For example, instead of speaking directly to a physician we would have another physician or a nurse together with us observing and explaining what another physician/nurse was doing. Doing so, we were able to interact with someone that was aware of what was going one without direct interfering with the process. One collateral and unexpected effect was that we could frequently hear critical remarks about the way the task was being performed. Since the person who was explaining the process was not the one doing the task he/she often felt more willing to tell us about problems. However one might be careful when considering these kind of remarks since sometimes the person pointing out the problem may be doing so simply because there are some kind of rivalry between them.

3.6. Video and Audio Transcripts

One important lesson learned in these years is that the use of video and audio transcripts can be very difficult. Due to legal concerns, organizations in this domain tend to deny the use of any video or tape recording during interviews as well as in the working areas. There is a major concern with the possibility that by any chance one might record something that could be used later to sue the organization and might be eventually summoned to hand over the tape to justice.

3.7. Use Cases and Scenarios

Use cases [16] and scenarios [17] were frequently used during the requirements elicitation. Although use cases have had its importance on showing the overall process, we frequently had to deal with details and exceptions that were more efficiently handled by scenarios. In general, the stakeholder felt more comfortable to validate scenarios descriptions than use case models.

3.8. Observation

Observation was occasionally used but much more to confirm some information gathered using other techniques. Being a very complex and specific domain, even in vocabulary, simply observing activities can be quite frustrating and non-productive. In the other hand,

sometimes one wants to check if what was understood from an interview or through documents reading, is really what happens in reality. For that purpose observation can be useful.

4. Conclusion

For the past six years the author has been conducting several case studies in the health care domain. During these six years the author has participated in the software development process within three different hospitals in Brazil, three different hospitals in Canada and four different clinical analysis laboratories in Brazil. All these experiences provided the opportunity to test many different elicitation techniques and adapt them when needed. Also provided ground for learning some important lessons on what to do and what not to do during the process of requirements elicitation

This paper presented the lessons learned from these experiences and aims to help requirements engineering that in the future be involved in developing software for this domains to avoid some of the usual pitfalls.

This paper also hopes to bring this important domain into closer discussion on the appropriated elicitation techniques and the most suitable requirements elicitation process(es) to be used.

5. References

[1] DeMarco, T. "Structured Analysis and System Specification" New York, Yourdon 1978.

[2] Yu, E. "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering" *in Proc. Of the 3rd IEEE Int. Symp. on Requirements Engineering*, 1997 pp:226-235.

[3] Goguem, J. and Linde, C. "Techniques for Requirements Elicitation*" First International Symposium on Requirements Engineering*, IEEE Computer Society Press, 1993 pp:152-164.

[4] D'Souza, D.F and Will, A.C. "Objects, Components and Frameworks With UML: The Catalysis Approach" Addison-Wesley 1999.

[5] Leite J.C.S.P. and Franco, A.P.M. "A Strategy for Conceptual Model Acquisition " in Proceedings of the *First IEEE International Symposium on Requirements Engineering, SanDiego, Ca, IEEE Computer Society Press,* pp 243-246 1993.

[6] Davenport, T.H. and Short, J.E. "The New Industrial Engineering: Information Technology and Business Process Redesign" *Sloan Management Review*, Summer, pp:11-27.

[7] The Guardian Angel Web Site - http://www.ga.org/ga

[8] Miksch, S., Cheng, K. and Hayes-Roth, B. "An Intelligent Assistant for Patient Health Care" in Proc. Of First International Conference on Autonomous Agents, California 1997.

[9] VanLansweerde, A. "Goal-Oriented Requirements Engineering: A Guided Tour" *Proc of the 5th IEEE Int. Symp. on Requirements Engineering*, 2001 pp:249-262.

[10] Mylopoulos,J. Chung, L., Yu, E. and Nixon, B.*, "*Representing and Using Non-functional Requirements: A Process-Oriented Approach*", IEEE Trans. on Software Eng*, 18(6), pp:483-497, June 1992.

[11] Cysneiros, L.M. and Leite, J.C.S.P. 'Integrating Non-Functional Requirements into data model" *4th International Symp. on Requirements Engineering* – June 1999.

[12] http://www.dcs.shef.ac.uk/~haris/

[13] Moser, C. and Kalton, G. "Survey Methods in Social Investigation" Gower, 1971.

[14] Andrews, D.C. "JAD: A Crucial Dimension for Rapid Applications Development" *Journal of Systems Management*, March 1991 pp:23-31.

[15] Ericsson, K.A. and Simon, H.A. "Protocol Analysis: Verbal Report as Data" MIT, 1984.

[16] Fowler, M. and Scott, K. "UML Distilled" Addison-Wesley Inc, 1997.

[17] Leite, J.C.S.P. et.al. "Enhancing a Requirements Baseline with Scenarios." *Requirements Engineering Journal,* 2(4):184-198, 1997.