Abstract: Today that requirements engineer has at his or her disposal a plethora of field tested approaches in which to elicit system requirements. In some specific domains, however, the generality of the methods prevents them from being taken to the word. The health care domain is one of such domains. Not only it is a complex domain with many subtleties but it also presents many political and legal issues that have to be taken into account. This work exposes some of the lessons learned in more than six years working in several hospitals and laboratories. In particular, this paper discusses some elicitation techniques that had to be adapted in order to comply with the constraints imposed by several peculiarities intrinsic to the health care domain. It also points out to some special considerations that must be taken into account when one chooses a requirements elicitation technique.

1. Introduction

As we face system complexity, the greater is the challenge in requirements engineering tasks. To deal with challenge, a constant stream of research on improving methodologies and techniques is under way. Requirements engineering research, however, like any other computer science field, is done in general, abstract manner, not concerning particular domains. The applicability and adaptation effort to particular situations is to be evaluated and undergone by developers, case by case.

Although this situation is understandable, it may not suffice when one deals with particularly complex and unknown domains such as the health care domain. This domain has been demanding the development of many a software. Critical software, needless to state, because they involve people’s lives. In the health care domain we deal with many constraints to elicit software requirements, constraints that range from the availability of stakeholders to the restrictions on the use of some elicitation techniques due to legal issues. Some can be quite challenging to overcome, and if not correctly dealt with, may lead to an erroneous requirements specification.

Recently, the author has been noticing a growing number of Ph.D. thesis, mostly from Europe, that somehow use facilities in the health care domain. Reading these thesis and talking to some of the authors it was possible to realize that most 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 i* Framework [2], and techniques ranging from semi-structured interviews to ethnography. The paper presents many lessons regarding special cares one might have when using a methodology for eliciting requirements as well as adaptations to many elicitation techniques [3] used. 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 can be considered to quite probably apply to many of the organizations in this domain.

It is important to mention that this work is not supposed to tackle problems related to eliciting requirements for areas involved with bioinformatics like molecular biology, gene sequencing and gene expression. This is a completely different area with particularities that holds only for this kind of software and not in the health care domain as it is understood in this work.

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

2. Special Considerations to the Requirements Elicitation Process

This paper will not be worried about studying any individual methodology to evaluate whether this methodology is adequate or not to be used in the health care domain. It is understood that all of them have their strengths and their weakness. Hence, we will try to bring
light to some points that might deserve special attention regardless the methodology being used.

2.1. Understanding the Vocabulary Used in the Domain

The first problem one may face in the health care domain may be one of the most challenging one: to understand what the stakeholder is saying. Actually, the need for using some kind of vocabulary has also been acknowledge by the Catalysis approach [4]. 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 thus, capturing it in an organized form not only come in a low cost but also serves to facilitate the reuse of the knowledge about the domain. It has been particularly useful when new engineers join the development team during the software development process.

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. Worst than that, in this domain one single term frequently has more then 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 understand that to electronically sign a patient report means that once all the results are inputted to the patients report, the system will print this report with the physician signature. Another stakeholder sees the same term meaning that the system must have the ability of understanding that the test result is within a safe range allowing the system to automatically assign this result to the patient’s report. These are two different interpretations to the same term that could lead to quite different software requirements.

To tackle this problem is suggested to use any kind of glossary control. In particular the Language Extended Lexicon [5] was used in almost all the cases with good results. The objective of the LEL is to register the vocabulary of a given UofD. 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 symbols (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 interview 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 conducted with 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 are synonyms. 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.

2.2. Look For Different Ways of Doing Things

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. However, it is more likely to find softwares that were developed targeting only the automation of an existing process. The problem about only automating the existing problem is that the health care domain is one domain where Davenport’s ideas should strongly be applied. 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

---

1 “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.”
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 most important, the use of information technology can lead to a better and safer process.

Thus, it is important that the requirement engineer always search for new ways for achieving the same goals that are achieved today. It is very 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. Having this understanding about the domain can drive the requirements engineer to think in alternative and more efficient ways of achieving these goals. Take for example one situation that happened while the development of an information system for a clinical analysis laboratory was taking place. Analyzing the process, we noticed that the samples were tagged with labels that had the Patient Internal Number (PIN) printed on it. Trying to understand the reasons under this process we asked the stakeholders why the PIN was printed on the label. Most of them justified it saying this was the way of getting the samples identified by the analyzers (special equipments that process the tests) to recognize patient’s identification. Two of the stakeholders also pointed out that this was quite helpful for them when they needed to find a sample within the laboratory. We then asked why they needed to find a sample in the laboratory and the answer was that sometimes they have to rerun a test to confirm the result and there is no more material in the sample that was originally being used, but there are other samples in other areas of the laboratory that can be used for that and hence knowing where these samples are was quite important. We than asked if this process always worked fine or if there were any problems. They answered that indeed some times when a patient visited the laboratory more then once in a week could turn the process of finding a sample quite difficult, because in this case we would have two different samples with the same PIN and therefore there would be no way of knowing which one corresponds to each visit. The problem is that one cannot simply choose one of the samples, because different results can arises from samples drawn in different days. At the end, it was proposed that one of the requirements for the new system was to use a sample number that would identify one and only one sample. Also that the system must provide a way of scanning samples every time a sample moves from on place to another so one can knows where all the samples of a patient are at one particular moment.

Although this paper does not want to tackle any particular approach, we have to emphasize that to do this type of analysis an agent driven or a goal driven approach like i*[2] and KAOS [7] suites very well since one can model goals together with what the software must provide to satisfy these goals. 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 works have been showing the importance of non-functional requirements elicitation [8, 9, 10]. 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.

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. Of course it is not being advocated here that a software has to get zero defects. Although that is what everyone hopes to get, it is still only a wish. What is being advocated here is that the software has to be carefully tested and one must be always concerned about reliability aspects. 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

Figure 1 – Example of the LEL
were not available when an analyzer processes a sample since this analyzer would simply skip the patient.

Security is also very important. The requirements engineer must always have in mind 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 hematology tests. Only supervisors can input results for some tests when its 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 a patient report that comes out of a laboratory should not contain the name of the patient. Also when the information system is used to upload information to any equipment, patient’s name should always be omitted. 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 together with the HIV test name, it will not matter the result. 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.

Usability is of course, always a must. Many users in this domain are not experienced with computers and therefore usability has to be a major concern. Haris [11] 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, as health care facilities in these countries have been dealing with information technology for a while.

The author has found many situations where Usability requirements were quite differently from those one could expect. Because of the environment and the work routine, sometimes usability aspects falls into a complete different set of solutions from those that could be usually proposed. Just as an example, we have once considered providing one of the tasks that were performed within the a system for clinical analysis laboratory with the capability for scanning bar code labels, so one would not have to input 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 stakeholder some of them pointed out that this solution would be impossible for them, since there was no space available in the workbench to place a bar code reader. A pen was not an option because it frequently fails to scan the label and also because of the nature of the chemicals used in the bench that 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. Also check for conflicts among non-functional requirements. Although it applies to any domain, here non-functional requirements often conflict with at least another one.

Of course the requirements engineer needs a methodology to elicit requirements and several points were made regarding it above, but also elicitation techniques must be applied to gather information.

3. Elicitation Techniques

There are many elicitation techniques that have been used for a while. Each of them usually fits best for one situation than another and therefore, part of the requirements engineering challenge is to choose the right one for each moment. However, 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. Below are depicted 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. 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. Not to mention that 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 reasons could
be found to explain that, but in part it seems to be related to the fact that quite frequently nurses and physicians work in 12 or 24 hour shifts and frequently the initial hours of these 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 in all the five institutions the results were much better ranking 84% of questionnaires returned, almost all of them satisfactorily filled up. Hence, the use of questionnaires may be recommended only in an explorative phase with the upper management.

3.3. Interviews

Although structured interviews were used, the best result came from the use of open-ended interviews [3]. Actually, although based in the concepts of open-ended interviews, it is reasonable to say that our approach might be closer to conversation analysis [3]. The best approach was to prepare a list of topics that we wanted to address during the interview, but not 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 most of the 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 [12] were used much more to solve conflicts than to gather knowledge about the domain. 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 is reasonable to assume that health care domain is not very suitable to use JAD sections. One of the reasons to that might be the very competitive environment that is quite frequent to this domain, together with the ethical problems that underlies almost all conversations. In spite of these problems, we could get good results using JAD sections to deal with conflicting requirements that involved more than two stakeholders.

3.5. Protocol Analysis

Protocol Analysis [13] in its strict sense can be very inconvenient in the health care domain for several 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/her duties. Imagine a physician assessing a cardio-respiratory failure and having to talk to you what he/she 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.

What we have done was to introduce 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. This way 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 what was going one. Since the person who was explaining the process was not the one doing the job he/she often felt more willing to tell us about problems. Of course there was always the problem of whether the person pointing out the problem was doing that because there was indeed a problem or simply because there was 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 if not impossible. 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 subpoenaed to hand over the tape to justice.

3.7. Use Cases and Scenarios

Use cases [14] and scenarios [15] were also used during the requirements elicitation of many of the software we built in this domain. 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 used several times but much more to confirm some information gathered using other technique. As this domain is very complex and specific 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 efficiently used.

4. Conclusion

For the past six years the author has been conducting several case studies in health care domain. During this 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 laboratory 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 methodology(ies) to this domain.

5. References