Evaluating the Effectiveness of Using Catalogues to Elicit Non-Functional Requirements

Luiz Marcio Cysneiros
School of Information Technology
York University – Toronto – Canada
cysneiros@yorku.ca

Abstract: *Non-Functional Requirements (NFR) are subjective, interactive and relative, thus realizing the need for particular NFR is by itself a challenge. Furthermore understanding what the software must implement in order to cope with these needs may prove to be an even more challenging task. One way of addressing the need for help on NFR elicitation is the use of catalogues. However, it is not clear how effective it is to use them. This work investigates it through an empirical study where different teams will model the same problem. Two teams will use catalogues with a systematic method, another two teams will use catalogs in an ad hoc manner and yet another two teams will not use catalogues. We show at the end of this work that teams using catalogues performed significantly better.*

Keywords: Non-Functional Requirements, catalogues, i* framework

1- Introduction

Ineffectively dealing with NFR has led to a series of failures in software development [13], including the very well known case of the London Ambulance System [12] where the deactivation of the system right after its deployment was strongly influenced by NFRs non-compliance. Moreover, eliciting and modeling NFR is a challenge for any requirements engineer. Since NFR constrain a functionality of the system the need for this NFR is often blurred by the functionality itself. Aside from that, NFR are subjective, interactive and relative. Despite the growing interest among researchers on NFR, there is still a lack of processes complete and mature enough to help dealing with NFRs from the early stages of software development until deployment. Since software representation languages do not tackle non-functional concepts [11] NFR are hard to trace along the different phases and models of the software development process. Similarly, the rationale on how to cope with NFR needs is hard to trace

Furthermore, when the requirements engineer decides to *satisfice* (operationalize) a non-functional requirement he may cause conflicts with other non-functional requirements. Here we use the same notion used by Mylopoulos [14]: that an NFR can rarely be said to be satisfied. That is, treating NFRs as goals we bring to bear the notion of partial satisfaction. This notion led Hebert Simon to coin the term "satisfice" [16]. Goal satisficing suggests that the solution used is expected to satisfy within acceptable limits. If these conflicts do not arise or are dealt with during the software development process, they may result in a series of problems at implementation time. Furthermore, errors due to the improperly dealing with NFR are among the most difficult and expensive to fix [3],[10]. The identification and proper expression of NFRs are essential to the understanding and reasoning of the impacts of further design decisions. It is also important to keep the rationale involving the reasons the requirements engineer choose one alternative of satisficing a NFR instead of another.

Although the use of catalogues has been proposed to help eliciting NFR [6], [9], there are important questions about the use of such technique that are no clear yet. For example how one deals with the different levels of granularity to retrieve information? How NFR and their operationalizations will be structured within these catalogues, i.e. generalizations? Although those are important questions that have to be investigated, this work is not concerned with them. It is focused on investigating if the use of catalogues alone regardless the difficulties on using them can help eliciting NFR. The assumption here is that not infrequently, one imperfect solution is better then no solutions at all.

In order to test this hypothesis, controlled experiments were designed to study how different teams using catalogues in a systematic way would perform against other teams using catalogues in an ad hoc manner and teams not using catalogues.

This paper is structured as follows. Section 2 describes the experiment we carry out to evaluate the effectiveness of using catalogues, Section 3 illustrates

the systematic approach for using catalogues used by two teams, Section 4 discuss the results and Section 5 concludes the work pointing out limitations and future work intended to overcome some of these limitations.

2.0 – The Experiment

To evaluate how much the use of catalogues help or not to elicit NFR a controlled experiment was designed inspired by the project replication strategy as proposed in [2].

The control sample was created compiling many models generated by me, another researcher and one graduate student while developing other work [7], [8], [9] and [19]. The control experiment is based on the Guardian Angel Project [4]. A set of "guardian angel" (GA) software provides automated support to assess patients with chronic diseases such as diabetes or hypertension, integrating all health-related concerns, including medically relevant legal and financial information, about an individual. The exemplar builds on software agents representing the hospital (GA_Hospital), the family members at home (GA_Home) and the patient being monitored (GA_PDA). This personal system helps track, manage, and interpret the subject's health history, and offers advice to both patient and provider. The system maintains comprehensive, cumulative, correct, and coherent medical records, accessible in a timely manner as the subject moves through life, work assignments, and health care providers.

Since we used i* models in the work above, the teams had to model the GA using i* models. Two teams did it using a systematic approach to use catalogues for modeling NFR when using i* (Section 3). Two other teams modeled the GA also using catalogues but without following any systematic way. Finally, two other teams modeled the GA without using catalogues.

Six teams were assembled so that each one would be composed of two forth year undergraduate students. All students were trained in the ***i**** framework as well as on NFR modelling. All the teams had to develop ***i**** models for one specific problem. Three of them used the framework presented in this work and the other three did not. The teams had no idea other teams were working in the same case and were instructed not to comment the experiment with anyone else. All teams used the same problem description that can be found in [19]. Their solutions were evaluated against a control sample created as explained in the beginning of this section. For elicitation purposes I played the role of the stakeholder for the teams when they needed to solve conflicts but only by choosing alternatives never suggesting any possible operationalization.

The parameter used for evaluation was the number of operationalizations from the control experiment that were found by each team. We have also taken into consideration operationalizations they found that were not present in the control experiment. Of course the higher the number of operationalizations each team found the better was the team performance. We have also measured how long each team took (in hours) to finish the models in order to evaluate the overhead of using catalogues and therefore to further clarify how much they are effective to be used or not.

During the development students used catalogues from Chung's book [6], i.e. security, performance, accuracy and as well as catalogues I have been developing in the past years, more specifically on usability, traceability and privacy [5].

Although the ***i**** framework builds on NFR, due to the abstract nature of NFR initial models tend to contain few NFR if any. It is therefore necessary to improve these models with necessary NFR. During the past three years knowledge on satisfying NFR have been collected and stored in the form of catalogues using the ***i**** framework for this purpose. They contain several operationalizations for some NFR as well as the rationale on how these operationalizations were achieved. These catalogues also contain many possible conflicts with other NFR due to the use of these operationalizations. The content of these catalogues is the result from several different investigative work based on surveys on these NFR as well as practical use. We have also included findings due to applying these catalogues to several case studies, some of them in real life cases. Currently, there are four different catalogues capturing the rationale for operationalizing these NFR (Traceability, Usability, Privacy and Privacy for the Health Care domain). These catalogues play an important role in the framework presented in this paper and they can be found at [5]. Figure 1 shows part of one of these catalogues. The catalogue in question is for the operationalizations that may satisfice the Usability NFR. Usability is first decomposed into the subgoals: Usefulness, Ergonomics and Ease of learning. Figure 1 shows de Usefulness part.

These catalogues were built using ***i**** [20] constructs including: softgoals, goals, tasks, and beliefs. The softgoal concept is used in ***i**** to ex\press non-functional requirements. NFRs frequently interact with each other in complex ways. Qualitative reasoning can be carried out using contribution links among softgoals. The semantics of the links are based on the satisficing concept [6] introduced in section 1. The most common contribution types are Help/Hurt (positive/negative but not sufficient to meet the parental goal), Some+/Some- (positive/negative of unknown degree), whereas Make/Brake indicates positive/negative of sufficient degree. Although these distinctions are coarse grained, they are enough to help us decide whether we need further refinement and search for more specific softgoals and operationalizations. Contribution links allow one to decompose NFRs to the point that one can say that the operationalizations of this NFR have been reached (i.e., the goals are no longer "soft"). In fact, operationalizations can be viewed as functional requirements that have arisen from the need to meet NFRs.

Operationalizations are typically specified as tasks, each indicating a particular way of doing something. All the subcomponents of a task (refined using the task decomposition link (⫞) must be carried out. If there is more than one way to accomplish something, then the state of affairs to be achieved is represented as a goal with means-end links (⫯) linking to the alternatives.

Contribution links are the core of design decisions. By reasoning about how different operationalizations would contribute to satisfice a softgoal, one may decide which the best alternative to purse is. Based on the semantics of the contribution links [6], decision values are propagated from an offspring to its parents allowing one to visualize what impact would come from adopting one alternative over another

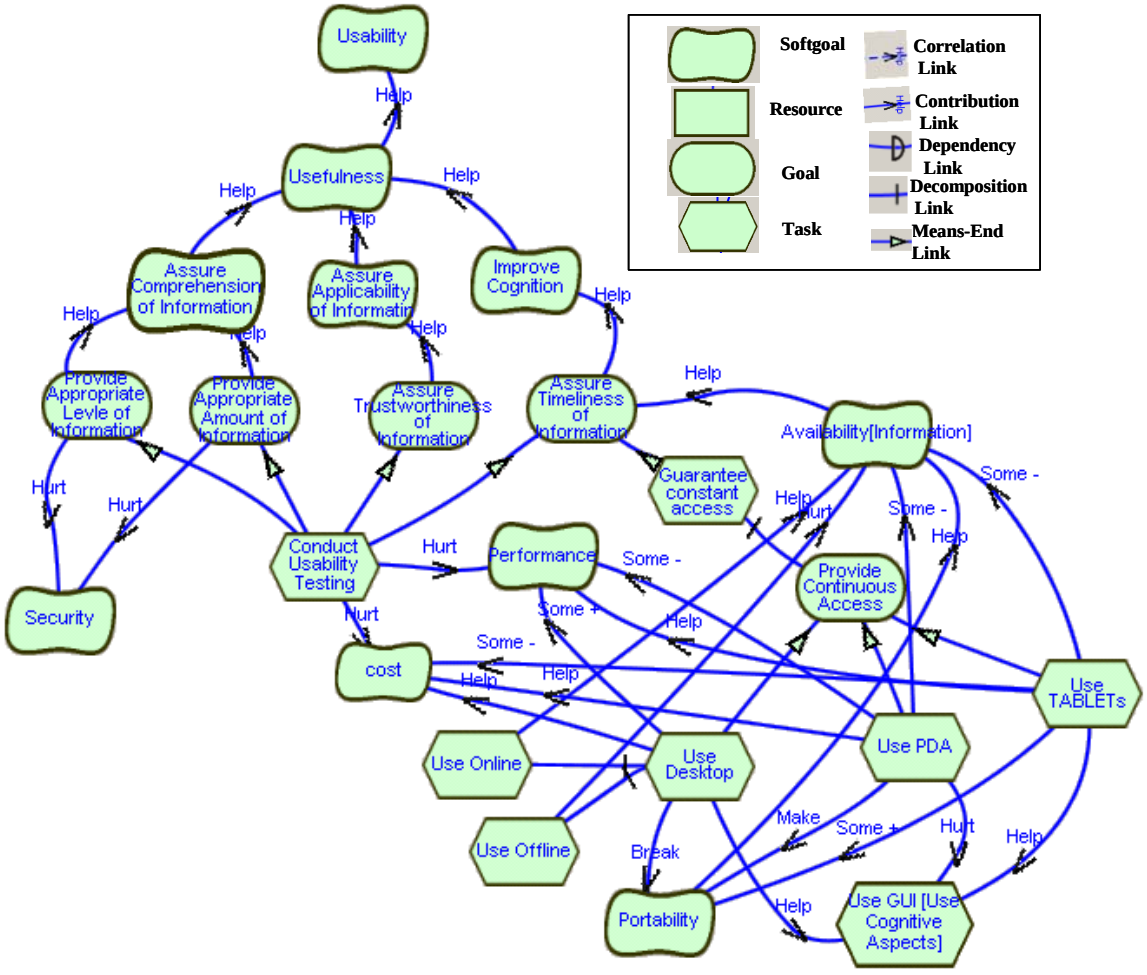

Figure 1 – Part of the Usability Catalogue

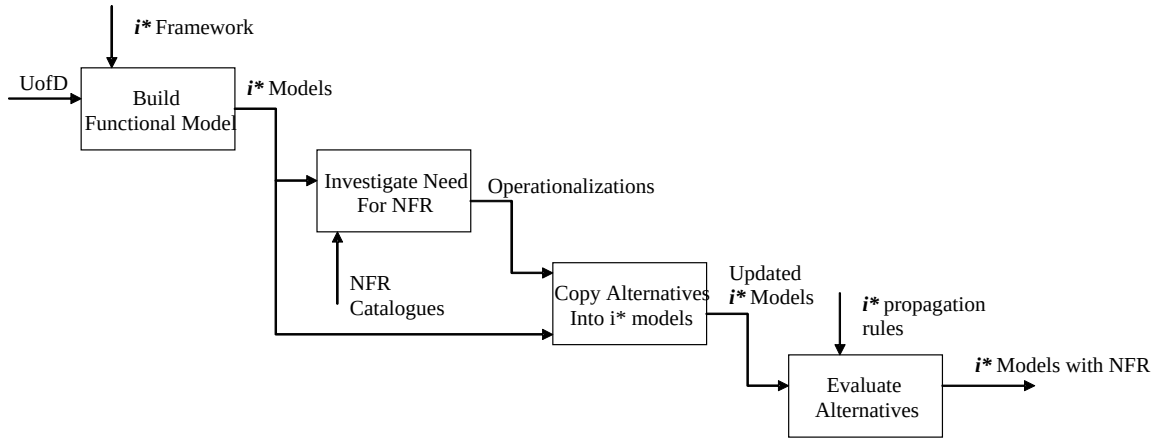

Figure 2 – A Systematic Approach to use Catalogue for Helping NFR Elicitation

3.0 – A Systematic Approach to use Catalogues to Elicit NFR

The first step is to build the functional models for the problem at hand. This follows the idea that typically we would address functional requirements first since they are more visible and easier to elicit. It does not mean, however, that functional models will be free of NFR. If the requirements engineer is able to elicit any NFR he/she should model them here, even though it is called "functional" models. Figure 2 shows a conceptual model for the framework

The second step is to enrich functional models with needed NFR. For doing so, we use the set of NFR catalogues available plus the list of generic NFR found in [6]. We start by asking ourselves and later the stakeholders whether any of these NFR could be important for any functionality in the models. If any NFR is needed, we select the possible alternatives for operationalization that would be applicable in this case.

The next step is to copy all alternatives into the functional models. Here the requirements engineer can choose among copying the complete rationale present in the catalogues, copying only part of the rationale, or copying only the operationalizations. While copying the rationale helps future reasoning on possible tradeoffs, it might turn the functional models into ones too loaded models hard to read. Since the rationale is kept in the catalogues anyway, one common alternative has been to use only one or two subgoals used to reach the operationalizations and if further rationale is needed we refer to the catalogues. The requirements engineering must also look at these catalogues to investigate if the chosen operationalizations may conflict with other NFRs. If there are any possible conflicts they should also be copied into the existing functional models for further evaluation in the next step of the framework.

Finally, the requirements engineering should first investigate if any conflicts will arise from bringing the chosen operationalizations into the existing models. All possible conflicts must be modeled. Then using the ***i**** propagation rules, the requirements engineer should investigate which of the chosen alternatives will better fit the solution domain. Possible new conflicts and alternatives arising from negotiation among different NFR satisficing should also be investigated and modeled.

3.1 - Illustrating the Framework

To illustrate the framework we will use material from one of the case studies carried out for building the usability catalogue [9]. The aim of the Surgery Control System is to control the planning and scheduling of surgeries in the Pedro Hernesto University Hospital (HUPES) in Rio de Janeiro, Brazil. The hospital needs to optimise and administer the surgeries to give better treatment in a Public Hospital.

The patient is sent to the surgery area after a physician's examination from HUPES or other Public Hospitals. Then the patient fills out a form with personal data and some generic symptoms and disease history data. Pre-surgery procedures consist of the patient's general exams like blood and urine analysis, as well as more detailed exams when needed such as CAT scans or MRI's for evaluating the patient's

surgical condition risk. An anesthetist analyses the patient's exams and assess patient surgical risk.

The head surgeon wants a system offering special features to help surgeons to better use their time, (for example by allowing them to schedule their own surgeries, except for high risk operations) or by allowing them to stay at home or at the office when they do not have a surgery to perform, while at other times to be on call for special occasions like emergency surgery. Some surgeons have a mobile phone others have a computer at home. Other surgeons do not frequently use the computer, and some have never used computers, so the system has to deal with different kinds of users.

The system should also allow the patient to provide his preference for the day of week and time and to access the system at home to confirm the surgery date. The patient should be able to see his data, prescriptions and nutrition restrictions. The first step according to the framework is to build the functional model for the problem. In our case we used the ***i**** framework. Figure 3 shows part of the ***i**** model to address the above scenario. Note that as said in section 2, although this is a functional model the NFR Usable Health Care system can be seen there. Due to the profile of the doctors using this system a usable system was a must and appeared quite early during the elicitation process.

The next step is to search for further NFR. For doing so we used the catalogues mentioned in section 2 plus the list of NFR that can be found in [Chung 00]. Here we found that for satisficing the Usable Health Care System NFR we needed Usability, Availability and Portability. We also realized that the task Set a New Surgery would need to satisfice Privacy concerns. At this point we checked the catalogues searching for possible alternatives for satisficing these NFR. Note that in this case we only used catalogues for Privacy and Usability. Due to space constraints this paper will focus on the Usability NFR. The software in question is expected to be heavily used inside the HUPES and also outside for the surgeons to consult and schedule their surgeries. It is also expected that the system would send messages to surgeons' mobile phones (for those who have one) so the surgeons can connect to the

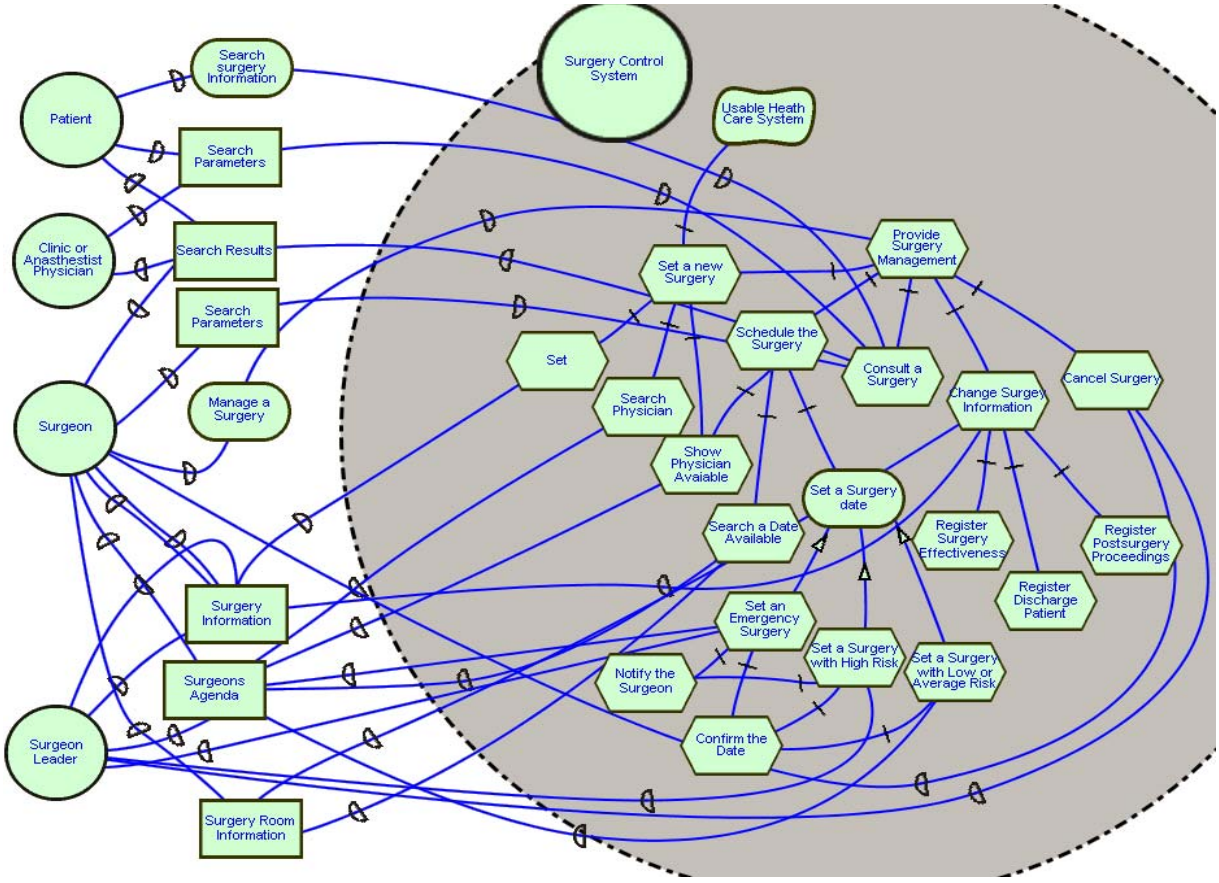

Figure 3 – The Functional Model for the Surgery Control System

system to inform their availability to perform an emergency surgery. Therefore, Usefulness is a major concern. To assess that, we determined that among the possible ways of satisfying Usefulness, Improve Cognition would be required leading to a further refinement for Assure Timeliness of Information. To satisfice that, we should Provide Continuous Access. This can be achieved by Using a Desktop, PDA or Tablets. However, from the catalogue we see that the use of a desktop would compromise Availability of Information (hurt correlation link) because for many applications as in the health care or in the retail industry, the need to reach a desktop in order to enter or consult data can be highly inconvenient. On the other hand, Using a PDA or Using tablets contributes positively (help correlation link) to Availability of Information. Figure 4 illustrate this step. It is time now to evaluate the different alternatives and make the necessary tradeoffs. We can see that to Use Desktop would help cost as well as it would have a positive contribution towards performance although in a lower level. It would also hurt Availability and break Portability. Use of PDA in its turn would help Cost and Availability while making a negative contribution towards Performance although in a lower level. It does, in fact, make portability while it also helps availability of information. As for Use TABLET, it would help Portability and Availability while hurting Cost. As for performance it is assumed it has a positive contribution towards it. Although the USE Desktop would make an important contribution towards Cost and Performance, two critical NFR in public hospitals, the fact it hurts Availability and Portability, which were considered much more important to a successful project, led us to decide not to Use Desktop. To Use TABLET implies in costs that could jeopardize project feasibility, furthermore since many doctors already have mobile phones that are also PDA we decided that at least for starting the project we would choose to Use PDA. Figure 5 illustrate this reasoning. Note that with the above scenario although we do not satisfice Cost and Performance we satisfice Usability, Portability and availability.

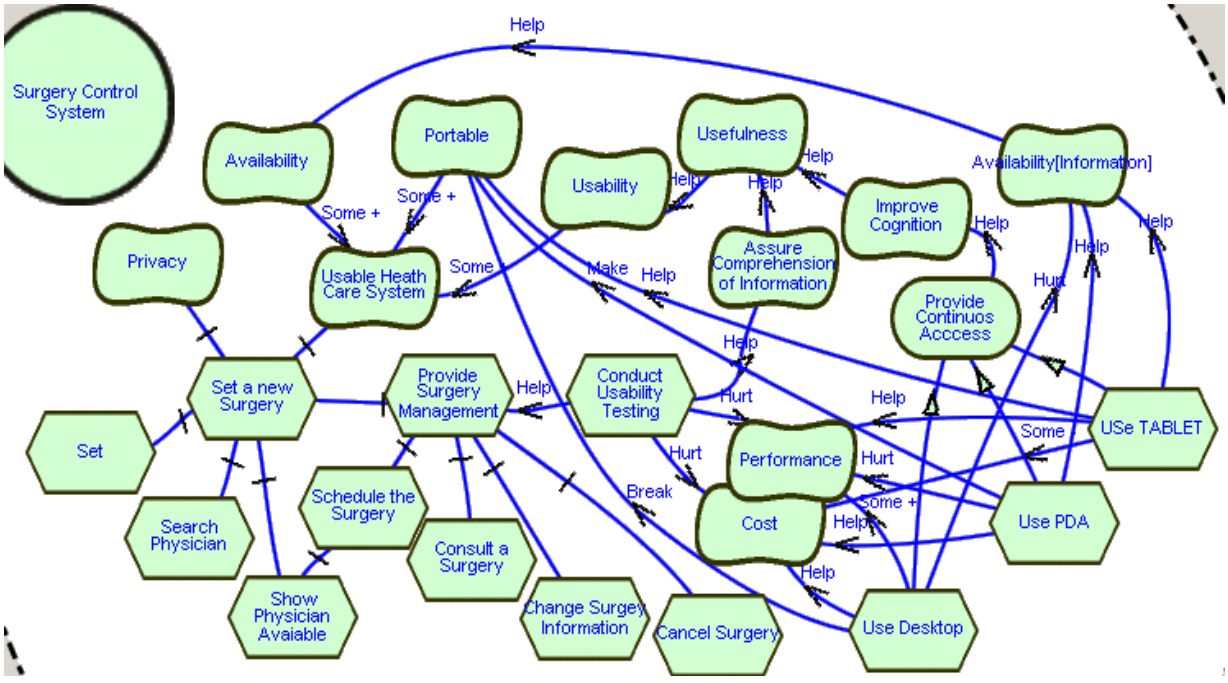

Figure 4 – Bringing Operationalizations into the model

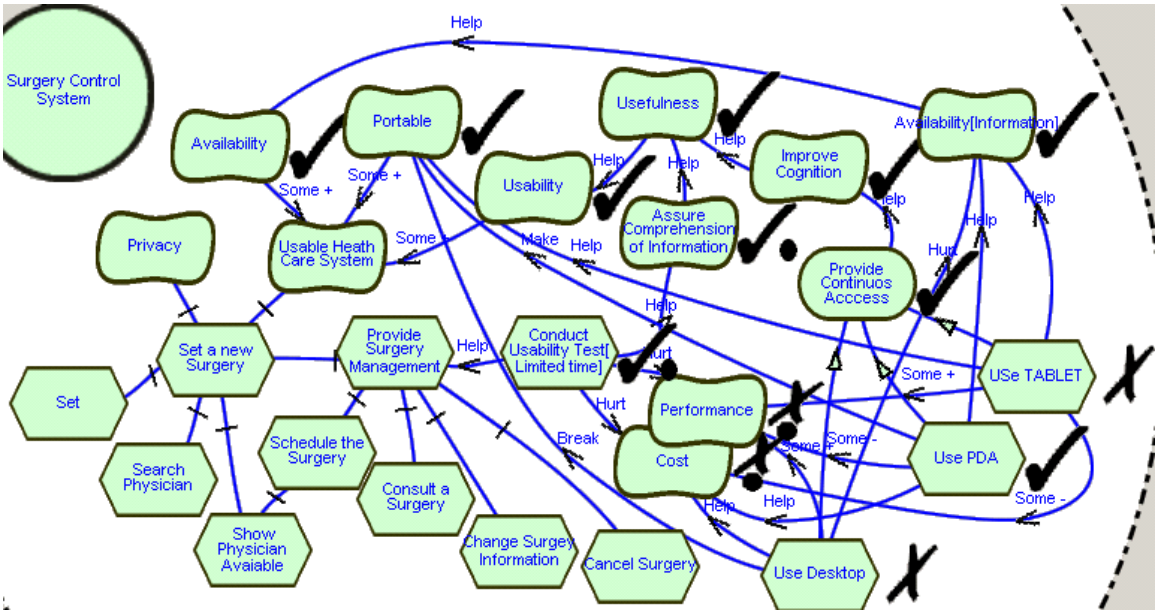

Figure 5 – Evaluating Alternatives

4 – Results from the Controlled Experiment

Table 1 summarizes the findings in terms of percentage of matching operationalizations and percentage of new operationalizations. Note that all six teams had the same training and were all composed of "A" students. The results suggest that the use of catalogues to elicit NFR was the most effective approach, since all teams using the framework found significantly higher numbers of operationlaization. One may suggest that the number of replicating projects is not enough to assure us the results are an indication of the helpfulness of the framework. However, although if one hand it is true that a larger number of groups would be better, on the other hand it is a challenge to find a larger number of students with similar backgrounds including training in ***i**** and NFR. Note that since all students are undergraduate students of an information technology program taking the same courses, and all of them with similar experiences in software development, we can expect the results to be only marginally influenced by individual capabilities. It seems particularly interesting that none of the teams *not* using catalogues had performed better or even close to the worst team

	Operationalizations Found (%)	New Operationalizations (%)	Hours to develop
Using Catalogues with a method			
Team 1	76	1	76
Team 2	88	2	74
Using Catalogues with no method			
Team 3	68	0	82
Team 4	61	1	70
Not Using Catalogues			
Team 5	31	0	57
Team 6	39	0	60

Table 1 – Results from the Experiments

using catalogues. This seems to be a clear indication of the usefulness of catalogues.

It is also possible to note that the teams using the systematic approach also performed better then the ones using an ad hoc approach. That seems to indicate that the use of systematic approach such as the one presented in section 3 augments the chances for eliciting a good set of NFR. However, it should be noticed that although the teams using the systematic approach performed better, the difference is not huge, which suggests that the use of catalogues may have been the key factor for all teams using catalogues performing better than those not using catalogues.

It is interesting to note that although we expected the teams using the systematic approach to expend more hours to finish the project, this was not entirely true. One of the teams using catalogues in an ad hoc manner has even spent more time than those using the systematic approach. This may be an indication that the used method may have a lower overhead than what was previously expected

There was one particular situation I was very interested in checking. It was regarding two conflicting NFR, Privacy and Availability. For the Guardian Angel Project, Privacy is very important because the patient medical history will be stored in the PDA. However, the PDA will be constantly connecting to other health agents through many different communications systems such as wireless network or internet. Thus, assuring medical records stay private is a major issue. On the other hand, availability of information is a must. The PDA should be able to access external medical databases in other to assess patient's health. It should also contain the medical records for providing easy transportation of medical records through life, work assignments, and health care providers. Many tradeoffs should be considered here such as: only storing medical records in the GA_Home and GA_hospital but that might implicate that transporting medical records would be a challenge. There is also the question whether we want the PDA to be password protected. While it helps on satisficing Privacy concerns it may hurt the availability of medical records in case of an emergency. As it can be seen from this brief discussion, there were many different operationalizations to be considered with many different outcomes for each possible alternative.

The teams using catalogues were able to tackle most of these alternatives. Here again, the teams using the systematic approach performed better than the ones not using it. On the other hand, the teams not using catalogues were not able to figure out any scenario involving both NFR. At best they realized the need for Privacy. In fact, examining the models produced by the teams not using catalogues we can see they failed to identify most of the conflicts. This should explain in part why they took much less hours to finish the project.

This particular result encourages the use of catalogues as well as to further investigate other ways of representing this knowledge. The way it is stored today may have been one of the reasons why one of the teams have not figure out the need for many of the operationalizations.

In fact, one interesting, yet expected, feedback I received from the teams using catalogues was related to the level of granularity to retrieve knowledge from the catalogues. Although addressing NFR from the more abstract level has often been used by all the teams, sometimes they have benefited from looking straight at the operationalizations. This feedback reinforce the need for further investigation on how we can store NFR knowledge to allow retrieving information using different levels of granularity.

Although it is clear that catalogues, the way they are presented here are not ideal the results from the experiments indicates that, in the lack of a better way for capturing this knowledge, catalogues, can be of great help for requirements engineers.

5 - Conclusion

This paper presents a framework for reusing knowledge on satisficing NFRs for goal-oriented approaches. The framework is centered on the use of catalogues that capture knowledge on satisficing NFRs captured based on existing literature [6] as well as on the experience acquired while conducting several case studies in other works, some of them in real life cases [7],[8],[9], [18]. It presents a systematic approach to use these catalogues for helping modelling problems using the ***i**** framework. It illustrates the use of the framework using one real life case study and evaluates its use through performing an in vitro controlled experiment. The results of this experiment suggest that the use of the framework can lead to models that can more comprehensively represent NFR needs.

Although this work only tackles the ***i**** framework I believe that other approaches such as KAOS [17] and GBRAM [1] could also benefit from this framework. In fact, future works will experiment the framework with the above- mentioned approaches.

Future work will also include a larger experiment as well as expanding this framework to be used by UML [15] developers.

Finally, I will also investigate different levels of granularity for reusing the knowledge stored in the catalogues. This will involve the development of a tool to be available in a web site that will allow people to access the catalogues retrieving the information stored. Different forms for representing these knowledge will also be investigated either using relational databases or ontologies.

Acknowledgements: This work is supported in part by NSERC grant number:262148-05

6 - References

[1] Antón,A.I. "Goal-Based Requirements Analysis," Second IEEE International Conference on Requirements Engineering (ICRE '96), Colorado Springs, Colorado, pp. 136-144, 15-18 April 1996.

[2] Basili, V.R. Selby, R.W., Hutchens, D.H. "*Experimentation in Software Engineering"* IEEE transactions on Software Engineering Vol. SE-12 No. 7 July 1986 pp733-742.

[3] Brooks Jr.,F.P.*"No Silver Bullet: Essences and Accidents of Software Engineering"* IEEE Computer Apr 1987, No 4 pp:10-19, 1987.

[4] P. Szolovits, Doyle, J., Long, W.J., "Guardian Angel: Patient-Centered Health Information Systems," http://www.ga.org/ga/manifesto/GAtr.html"

[5] http://www.math.yorku.ca/~cysneiro/nfrs/nfrs.htm

[6] Chung, L., Nixon, B., Yu, E. and Mylopoulos,J. "Non-Functional Requirements in Software Engineering" Kluwer Academic Publishers 2000.

[7] Cysneiros,L.M., Werneck, V. and Yu,E. *"Evaluating Methodologies: A Requirements Engineering Approach Through the Use of an Exemplar* "Journal of Computer Science and Technology Special Issue on Software Requirements Engineering Vol. 5 - No. 2 - July 2005

[8] Cysneiros,L.M., Werneck, V. and Yu,E. "Agent/goal Orientation versus Object Orientation for Requirements Engineering: A Practical Evaluation Using an Exemplar" in Proc. of VIII Workshop in Requirements Engineering, 2005, Porto, Portugal, pp:123:134

[9] Cysneiros,L.M. Werneck,V. and Kushirk,A. "Reusable Knowledge for Satisficing Usability Requirements" in the Proceedings of the 13th IEEE International Requirements Engineering Conference – Paris – August 2005.

[10] Davis, A. *"Software Requirements: Objects Functions and States"* Prentice Hall, 1993.

[11] Feather, M.S., Menzies,T., Connelly,J.R. "Relating Practitioner Needs to Research Activities" in the the Proceedings of the 11th IEEE International Requirements Engineering conference – Paris – September 2005, pp:352-361.

[12] Finkelstein, A. and Dowell J. "*A comedy of Errors: The London Ambulance Service Case Study*" Proceedings of the Eighth International Workshop on Software Specification and Design, IEEE Computer Society Press pp 2-5 1996.

[13] Lindstrom, D.R. "*Five Ways to Destroy a Development Project"* IEEE Software, September 1993, pp. 55-58

[14] 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

[15] J. Rumbaugh, I. Jacobson, and G. Booch, The Unified Modeling Language Reference Manual: Addison-Wesley, 1999.

[16] Simon,H.A. "The Sciences of the Artificial" , 3rd edition MIT Press, 1996.

[17] Van Lamsweerde, A "Goal-Oriented Requirements Engineering: A Guided Tour," Proc. of 5th IEEE Int. Symp. on Requirements Engineering, 2001

[18] Webster,I., Ivanova,V., Cysneiros,L.M. "Reusable Knowledge for Achieving Privacy: A Canadian Health Information Technologies Perspective" in Proc. of VIII Workshop in Requirements Engineering, 2005, Porto, Portugal,pp:112:122

[19] Yu,E., Cysneiros.L.M., "Agent-Oriented Methodologies-Towards a Challenge Exemplar" Wagner, G.; Karlapalem, K.; Lesperance, Y.; Yu. E. (Eds.) Agent-Oriented Information Systems 2001.

[20] 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.