Dealing with Non-functional Requirements, the Case of Information Quality Requirements: Experience Report

Mohamad Gharib

University of Florence - DiMaI, Viale Morgagni 65, Florence, Italy
mohamad.gharib@unifi.it

Abstract. Like all Non-functional Requirements (NFRs), Information Quality (IQ) requirements are used to be represented as softgoals that are difficult to be represented measurably. However, several recent studies argued that many requirements that are classified as NFRs can be expressed in a measurable way. This paper reports on experience gained while proposing a goal-based approach for capturing IQ requirements as softgoals at a high-level of abstraction and then refining them until reaching their operational specifications.

Keywords: NFR, Softgoals, IQ, Requirements engineering

1 Introduction

Generally speaking, Non-functional Requirements (NFRs) are difficult to be expressed in a measurable way since they do not have clear-cut criteria for satisfaction [1]. Unsurprisingly, Information Quality (IQ) requirements, like all NFRs, are used to be represented as softgoals without specific methods for their analysis. However, several authors [2,3] argued that many requirements that are classified as NFRs can be handled similarly to Functional Requirements (FRs), i.e., they can be associated with clear criteria for their satisfaction. Despite this, very few approaches have been proposed for providing such criteria (e.g., [3]).

The main objective of this research is proposing a goal-based approach for capturing one type of NFR, namely IQ requirements as softgoals at a high-level of abstraction. Then, refining them until reaching their operational specifications. The posed *Research Questions* are: *RQ1:* How can we model IQ requirements? *RQ2:* How can we refine IQ requirements until reaching their operational specifications? and *RQ3:* How can we verify the correctness of such models?

A case study concerning a main stock market crash[1] has been used to illustrate this approach [5]. In particular, the main factors that contributed to this crash were due to IQ related vulnerabilities. For instance, some traders intentionally provided inaccurate (untrustworthy/unbelievable) information to manipulate the trading environment. Yet, such vulnerabilities could be avoided if the IQ requirements of the system were properly captured during system design.

[1] For more information about the case study please refer to [4]

The rest of this report is structured as follows; Section 2 presents the approach, and its implementation and evaluation are discussed in Section 3. Section 4 discusses the findings, and the report is concluded in Section 5.

2 Approach for Specifying IQ Requirements

The process that underlies the goal-based approach for specifying IQ requirements is depicted in Figure 1, and it consists of two main phases:

1. Modeling phase aims at modeling the IQ requirements in their social and organizational context. This phase consist of eight main steps: (1.1) *Actor modeling,* models the main actors of the system in terms of agents and the roles they are playing; (1.2) *Goal modeling,* models the actors' objectives in terms of goals, and refine them until reaching their leaf goals; (1.3) *Information modeling,* models the different relations between goals and information; (1.4) *Social dependency modeling,* models actor dependencies for information, and the delegations of permissions and goals; (1.5) *Trust modeling,* models trust/distrust relations among actors concerning their social dependencies.

The following three steps (S.1, S.2 and S.3) are specialized for dealing with IQ softgoals, and they can start after (1.3) *Information modeling* step:

(S.1) *IQ softgoal modeling* aims at modeling top-level IQ softgoals and refines them into a suitable granularity that enables for their approximation. Usually, softgoals refinement into more specific sub softgoals can be done based on taxonomy (e.g., [1]). Thus, the first step is defining an *IQ refinement taxonomy.*

Define IQ refinement taxonomy. The literature is rich with models (taxonomies) for analyzing IQ based on various dimensions, but most of them were not designed to capture the social and organizational aspects that underlie some of IQ dimensions. Therefore, a multi-dimensional model (Figure 2) for analyzing IQ based on seven dimensions has been developed [6]: *1. Accessibility* captures the extent to which information is available for use. *2. Believability* captures the extent to which information is regarded as true. *3. Trustworthiness* captures the extent to which information is credible, and it is analyzed based on *trustworthiness of the source* and *trustworthiness of the provision. 4. Accuracy* captures the extent to which information is error-free with respect to some known value. *Accuracy* is analyzed based on *believability* and *trustworthiness. 5. Completeness* captures the extent to which information is complete for performing a specific task. *Completeness* is analyzed depending on two sub-dimensions, *value completeness*, and *purpose of use completeness. 6. Timeliness* captures the extent to which information is valid in terms of time. *7. Consistency* captures the extent to which multiple records of the same information are the same across time. These dimensions have been considered based on the requirements of the stock market system, and they might need to be extended or reduced for other systems.

Following [7], we can depend on this taxonomy to refine IQ softgoals into IQ sub-softgoals through *and-decomposition* relations, since a softgoal can be refined into more specific sub-softgoals, if the joint satisfaction of these softgoals is considered equivalent to the satisfaction of the refined softgoal. Figure 3 shows

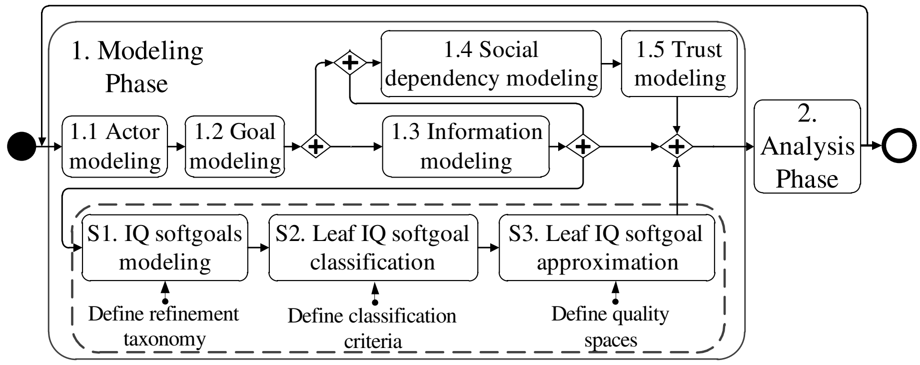

Fig. 1. The process for specifying Information Quality requirements

the application of the taxonomy for refining a top-level IQ softgoal concerning information (e.g., "Trading order"). After refining all top-level IQ softgoals into their leaf IQ softgoal based on the taxonomy, we can proceed to the next step.

(S.2) *Leaf IQ softgoal classification* aims at identifying classification criteria to be followed while organizing leaf IQ softgoals into groups concerning their satisfaction criteria, since they capture different IQ dimensions (e.g., accuracy, completeness) and they might need to be approximated in different ways.
Define classification criteria. Following Glinz [2], leaf IQ softgoals have been classified based on their kind, satisfaction and representation to get a better understanding of their nature and how they can be approximated. Table 1 shows how leaf IQ softgoals have been classified based on these criteria. After classifying leaf IQ softgoals, we can proceed to the next step.

(S.3) *Leaf IQ softgoal approximation* aims at approximating leaf IQ softgoals into IQ Constraints (IQCs). In particular, a softgoal can be satisfied by a quality constraint through the *approximation* relation [7], where a quality constraint pro-

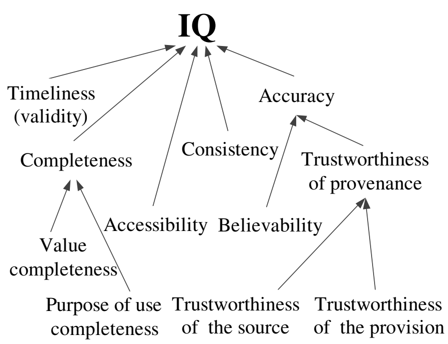

Fig. 2. A Multi-dimensional Model (taxonomy) for Analyzing IQ

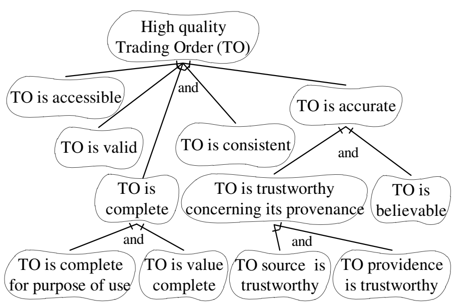

Fig. 3. The application of the model to the IQ of "Trading order"

Table 1. IQ softgoal classification & approximation into IQC

Leaf IQ softgoals	**Kind**	**Satisfaction**	**Representation**	**Approximated into IQC**
Believability	Functional	Hard	Operational	Operational IQC
Trustworthiness	Constraint	Hard	Declarative	Declarative IQC
Completeness	Constraint	Hard	Declarative	Declarative IQC
Timeliness	Performance	Hard	Quantitative	Quantitative IQC
Consistency	Performance	Hard	Quantitative	Quantitative IQC

vides clear criteria for the satisfaction of a softgoal. However, an approximation relation can hold only if a well-defined quality space exists [7].

Define quality spaces. IQ dimensions must be associated with specific measures (quality space) to assure their effective assessment. To this end, a quality space for each IQ dimension has been defined based on the following criteria: the measurement should be easy to be interpreted, the difference between the value levels of measurement must be meaningful, and the calculation of such values should be consistent. For instance, timeliness and consistency are time-related aspects; therefore, they are measured and calculated in seconds. How IQ dimensions are analyzed should be clearly defined as well, e.g., *validity* is analyzed by comparing the currency (seconds) of information with its volatility (seconds), and if its currency is smaller than its volatility it is valid, otherwise, it is invalid.

After defining measures for analyzing leaf IQ softgoals, we can approximate them into IQCs. Three types of IQCs have been defined (shown in Table 1): (1) *Operational IQC* define actions to be performed in an already determined context. E.g., *believability* softgoal is approximated into operational IQC that define min/max values concerning information believability. (2) *Declarative IQC* define properties of the system that should hold. E.g., *trustworthiness of provision* softgoal is approximated into declarative IQC stating that information should be transferred only through IP provision. (3) *Quantitative IQC* specify properties of the system that should hold, and can be measured on an ordinal scale. E.g., *consistency* softgoal is approximated into quantitative IQC stating that information should have the same currency among its *interdependent readers* that are actors who use the same information for interdependent purposes.

Figure 4 shows a portion of the stock market system model represented with an extended modeling language [5,6]. The language adopts several main *i** based constructs such as *actor*, *goal*, *delegation*, *trust*, etc., it extends some *i** constructs and propose new constructs specialized for IQ requirements. For instance, *goals* may *produce*, *read*, *modify* and/or *send information*. *Information Provision* has a transmission time attribute, and a provision type attribute that can be either Integrity-Preserving (IP) or normal Provision (P). *IQ softgoal* is an objective of a stakeholder concerning its needs over information, and it can be refined only through *and-decomposition*. IQCs have a well-defined quality space, and they can be used as a mean to satisfy IQ softgoals through *approximation* relation.

The language relies on four types of permissions for capturing *accessibility*. Moreover, *read* and *produce* relationships have been extended with believability checks to capture *believability*. *Trustworthiness* is analyzed based on the *trust-*

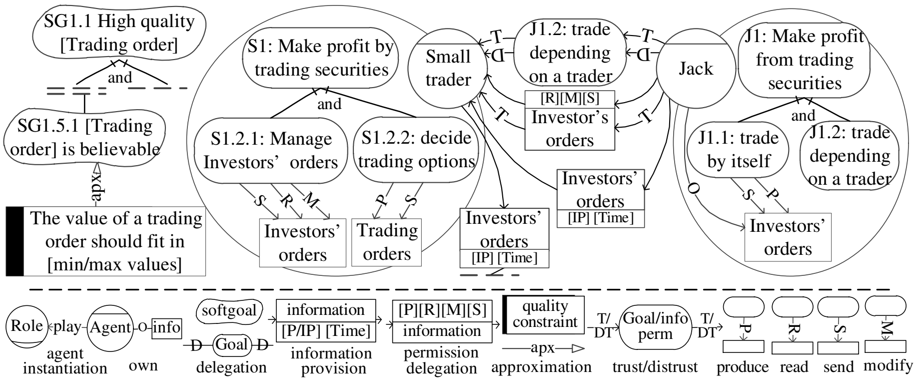

Fig. 4. A partial goal model concerning the stock market structure

worthiness of both the *source* and the *provision. Accuracy* is captured relying on both *believability* and *trustworthiness. Completeness* is analyzed based on the provision type and the *part of* concept that captures the relationship between information and its sub-parts. The language proposes *information volatility*, *read* and *send timeliness* concepts for capturing *timeliness.* Finally, it provides *interdependent readers* and *read time* concepts for analyzing *consistency.*

2. Analysis phase. To verify the IQ requirements model, all the proposed concepts have been formalized, and reasoning axioms have been developed relying on Disjunctive Datalog. Moreover, a set of Properties of the Design (PoD) have been defined. The model is considered correct and consistent if all of the PoD hold. If any of them is violated (e.g., information is inaccurate, incomplete, inconsistent, etc.), the designer is notified about that, which allows her to modify the model to address such violation.

3 Implementation and evaluation

The approach has been evaluated depending on a simulation method, developing a prototype implementation and tests its applicability by applying it to the Flash Crash case study. The approach was able to models and effectively analyze (e.g., detecting any violation to the properties of the design) the IQ requirements of the case study. Moreover, the scalability of the reasoning support of the approach has been evaluated, and it was able to deal with sufficiently large models[2].

4 Discussion

Most software engineers need to deal with NFRs, yet some of these NFRs can be responsible for providing essential functionality for the system such as security,

[2] For more information about the implementation and evaluation please refer to [6]

privacy, reliability, etc. Therefore, they cannot be dealt with vaguely, i.e., clear criteria for their satisfaction should be provided. Such criteria will not only facilitate the system design, but they will also help both stakeholders and software engineers to better understand each other, which may prevent wrong design decisions. The proposed approach has been successfully used in a framework for modeling and analyzing IQ requirements for a Socio-technical System [8]. Moreover, we mainly depend on it while dealing with privacy requirements (NFRs) in the VisiOn Project (`http://www.visioneuproject.eu`). In particular, we proposed a taxonomy for capturing and refining privacy requirements until reaching their operational specifications [9]. To this end, I believe that this approach can assist software engineers when dealing with a wide spectrum of NFRs.

5 Conclusions

This paper reports on experience gained while proposing a goal-based approach for capturing IQ requirements as softgoals at a high-level of abstraction and then refining them until reaching their operational specifications. The focus was put on making the approach easy to be understood and used, and each of its steps was accompanied by a detailed description of how it can be performed. This may help other scholars while dealing with NFRs.

References

1. Mylopoulos, J., Chung, L., Nixon, B.: Representing and using nonfunctional requirements: A process-oriented approach. IEEE Transactions on Software Engineering (1992) 483–497
2. Glinz, M.: On non-functional requirements. In: Requirements Engineering Conference, 2007. RE'07. 15th IEEE International, IEEE (2007) 21–26
3. Li, F.L., Horkoff, J., Mylopoulos, J., Guizzardi, R.S., Guizzardi, G., Borgida, A., Liu, L.: Non-functional requirements as qualities, with a spice of ontology. In: Requirements Engineering Conference (RE), IEEE (2014) 293–302
4. Gharib, M., Giorgini, P.: Detecting conflicts in information quality requirements: the May 6, 2010 Flash Crash. Tech. report, Universitá degli studi di Trento (2014)
5. Gharib, M., Giorgini, P.: Modeling and reasoning about information quality requirements. In: Requirements Engineering Foundation for Software Quality, Springer (2015) 49–64
6. Gharib, M., Giorgini, P.: Dealing with information quality requirements. In: International Conference on Enterprise, Business-Process and Information Systems Modeling, Springer (2015) 379–394
7. Jureta, I., Mylopoulos, J., Faulkner, S.: Revisiting the core ontology and problem in requirements engineering. In: RE Conference, IEEE, 2008 71–80
8. Gharib, M., Giorgini, P., "Information quality requirements engineering with STS-IQ," *Information and Software Technology*, vol. 107, pp. 83–100, 2019.
9. Gharib, M., Salnitri, M., Paja, E., Giorgini, P., Mouratidis, H., Pavlidis, M., Ruiz, J.F., Fernandez, S., Della Siria, A.: Privacy requirements: Findings and lessons learned in developing a privacy platform. In: the 24th International Requirements Engineering Conference (RE), IEEE (2016) 256–265