Representing and Elaborating Quality Requirements: The QRA Approach

Jie Sun[1], Pericles Loucopoulos[2], and Liping Zhao[3]

[1] College of Computer Science, Zhejiang University, Hangzhou, China
jiangbin@zju.edu.cn
[2] Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece
p.loucopoulos@hua.gr
[3] School of Computer Science, University of Manchester, Manchester, U.K
liping.zhao@manchester.ac.uk

Abstract. This paper presents the quality requirements analysis (*QRA*) approach to requirements modeling. The *QRA* approach supports functional and non-functional requirements modeling in three dimensions: First, it extends functional goal modeling with non-functional goals. Second, it injects QR (quality requirement) specification into business process models. Third, it provides a set of rules for elaborating and refining QRs alongside functional decomposition of business processes. This paper describes *QRA*'s conceptual foundations and illustrates them through goals and business processes of a real world stock trading system.

Keywords: quality requirements (QRs), goal modeling, business process modeling, quality requirements analysis.

1 Introduction

Although quality requirements (QRs) such as reliability, performance, security, compatibility are critically important to software systems, they have often been left out at the early requirements analysis stages and only added to the system at a late stage [1, 2]. According to a recent study [3], this problem might be caused by the significant involvement of system architects in early requirements elicitation, whose concerns are primarily functional requirements, rather than NFRs or QRs. Even when QRs are considered, they are often represented informally, using in-house templates and notations [1].

This paper presents the quality requirements analysis (*QRA*) approach to requirements modeling. The *QRA* approach supports functional and non-functional requirements modeling in three dimensions: First, it extends functional goal modeling with non-functional goals. Second, it injects QR (quality requirement) specification into business process models. Third, it provides a set of rules for elaborating and refining QRs alongside functional decomposition of business processes. This paper describes *QRA*'s conceptual foundations and illustrates them through goals and business processes of a real world stock trading system.

W. Ng, V.C. Storey, and J. Trujillo (Eds.): ER 2013, LNCS 8217, pp. 446–453, 2013.
© Springer-Verlag Berlin Heidelberg 2013

2 Conceptual Foundations of QRA

The QRA approach builds on the following three conceptual foundations:

- A *meta-model* that represents the concepts of QRA and their relationships.
- A *meta-process* defining the way of working with QRA.
- A set of *formulae* for elaborating the QRs.

Due to space limitation, this section will only describe the first two foundations. The description of the third foundation is given in a separate paper [4].

2.1 The QRA Meta-model

The meta-model of the QRA approach, shown in Fig. 1, defines a set of concepts for representing and elaborating QRs and FRs.

Requirements, in the QRA approach, are considered through a prism of two major conceptual artifacts namely *business goals* and *business processes*. These represent the basic building blocks for defining FRs and QRs. Since the QRA approach is primarily concerned with QRs, most of the details in the meta-model are about QRs but for completeness purposes, the meta-model also includes a part of FRs definition in order to show the interplay between QRs and FRs.

The QRA approach adopts a goal-oriented approach, which in recent years has become a dominant paradigm in RE [5]. Goal modeling involves a series of causal relationships that essentially analyze goals from fuzzy and vague [6] to goals of increasing determinism. Through these causal relationships, goals are organized into a goal hierarchy with the leaves referred to as "`operational goals`" and the others as "`intermediate goals`". There are two types of causal relationship, one that is concerned with the causality of type "`refine`" between intermediate goals, signifying how a goal, the "`satisficed goal`", is deconstructed into one or more "`satisfier goal(s)`", and one that is concerned with the causality between intermediate goals and operational goals, signifying how a goal is "`operationalized`".

Operational goals are business goals, i.e. customer-facing goals, with exact "`requirements`" for systems intended to satisfy these business goals. Requirements are distinguished into "`FRs`" and "`QRs`". In QRA, FRs and QRs are considered synergistically because there is an inevitable intertwining between them [7].

As shown in the meta-model of Fig. 1, FRs are defined as either the tuple of `<Transaction, Business Process, Function Point>` concerned with business processes at a macro level or `<Transaction, Business Process Element, Function Point>`, concerned with individual components of business processes. Both ternary relationships are depicted in the meta-model as the objectified element of "`Function Component`". QRs are defined symmetrically to FRs as either the tuple `<Quality Factor, Business Process, Quality Metric>` or `<Quality Factor, Business Process Element, Quality Metric>`, both of which are shown as the objectified element "`Quality Component`".

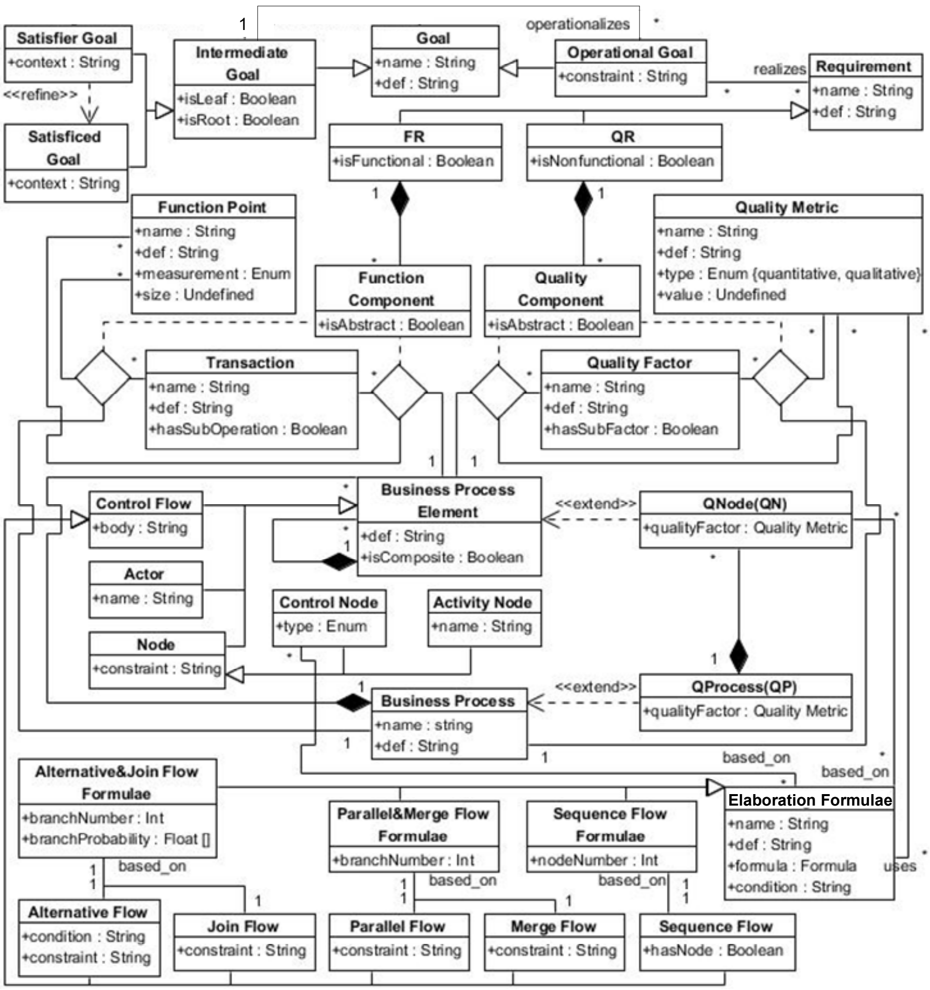

Fig. 1. The QRA meta-model

A business process is a set of ordered behaviors performed by actors to meet specific functional requirements. Generally, there are three types of business process element: i) `actors` that are responsible for performing behaviors; ii) `nodes` including `activity nodes` that specify behaviors and `control nodes` that coordinate execution flows of activity nodes; ii) `control flows` that define execution order of nodes and form the basic structure of each business process. There are five types of control flow, namely `Sequence Flow`, `Parallel Flow`, `Merge Flow`, `Alternative Flow and Join Flow` [8].

Since FRs and QRs are interrelated through business processes, we define an integrated representation of FRs and QRs through two new concepts, `QProcess (QP)` and `Qnode (QN)`, which extend `Business Process` and `Business Process Element`. A QP is a quality-enhanced business process composed of QNs. Each QN integrates a functional business process element with annotations of quality factors and metrics. Although any business process element can be annotated with QRs, due to space limitation, this paper focuses on the incorporation of QRs into activity nodes.

For each QP as a whole, a set of `Elaboration Formulae` are defined to measure its QRs, by aggregating the quality metrics annotated on its QNs. According to the five basic control flows of QPs, three types of formulae are defined, which are `sequence flow formulae`, `alternative&join flow formulae` and `parallel&merge flow formulae`. Quality metrics of the overall QP can be calculated by iteratively decomposing the QP into sub-QPs and delegating the calculation to them until each sub-QP can be matched to a basic control flow and calculated by correspondingly formulae.

2.2 The QRA Meta-process

The meta-process of the QRA approach is presented in Fig. 2 and illustrated in Section 3. The QRA meta-process defines the main steps of the approach that involve *goal modeling*, *business process modeling* and *quality requirements elaboration*, making use of the concepts defined in the meta-model of Fig. 1.

```
Begin
    Specify high-level customer goals
    While goal modeling is incomplete                    //Goal Modeling
        For each intermediate goal
            Define satisfycing relationships
            Specify satisfier goals
        EndFor
        Define operationalization relationships
        Specify operational goals
            For each operationalized goal
                Identify requirements
                Identify business processes corresponding to requirement
            EndFor
    EndWhile
    While business process modeling is incomplete        //Business Process Modeling
        For each requirement
            Identify FRs and QRs
            For each FR
                Specify function components
                Specify transactions
                Specify function points
            EndFor
            For each QR
                Specify quality components
                Specify quality factors
                Specify quality metrics
                Extend business process with QProcess
                Extend Business process element with QNode
            EndFor
        EndFor
    EndWhile
    While QR elaboration is incomplete                   //QRs elaboration
        Decompose a QP into sub-QPs
        For each sub-QP
            Calculate quality measures with elaboration formulae
        EndFor
        Calculate quality metrics for the overall QP with elaboration formulae
        Compare results with customer needs
    EndWhile
End
```

Fig. 2. The QRA meta-process

3 Illustrating the QRA Approach

The concepts introduced in section 2 are illustrated in this section through an example of a stock trading service system (STSS) from a US-based multinational financial firm. STSS provides real-time services to facilitate its users (i.e., traders) to sell or buy shares online and this is realised by interacting with two other agents – Market Data Feeder (MDF) and Printing Agent (PA). STSS needs to fulfill five essential FRs as well as a set of QRs. The five FRs are: (i) Order Entry, (ii) Order Pricing, (iii) Order Matching, (iv) Trade Printing and (v) Trade Allocation. Due to space limitation, this paper only illustrates three key QRs of the system, which are: (i) performance, (ii) recoverability and (iii) accessibility. The following subsections illustrate the three major phases outlined in the QRA meta-process of Fig. 2.

3.1 Goal Modeling

According to the QRA meta-process, a goal model was constructed to analyze high-level goals to operational ones and from those to define FRs and QRs. A small part of the STSS goal model is shown as Fig. 3. The notation used is an extension of two popular goal modeling techniques namely KAOS [9] and i* [10] and implemented in a prototype tool [11]. The high-level business goal - *to make the firm competitive in the market* - is decomposed into intermediate goals which through successive analysis are defined in terms of the operational goals *1.1.1.1* and *1.1.1.2*. In order to realize the two operational goals, we define realization relationships and specify nine requirements, five FRs and four QRs. For QRs, we further identify three relevant quality factors – *recoverability* (QR1), *accessibility* (QR2) and *performance* (QR3&QR4).

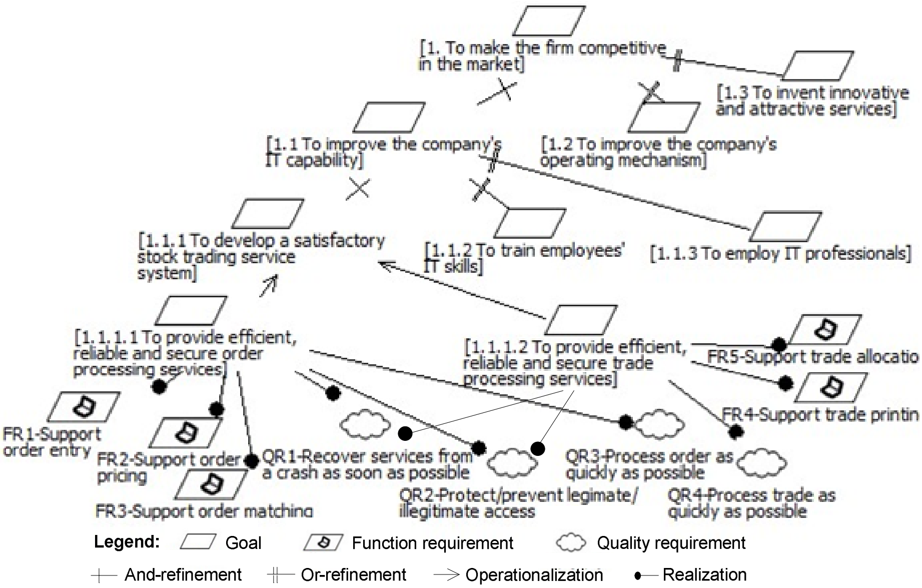

Fig. 3. The Goal Modeling Graph for STSS

3.2 Business Process Modeling

For this example, the business process with its corresponding transactions is shown in Fig. 4 using UML Activity Diagram notation. However, this notation is augmented in order to incorporate the extensions to QProcess and QNode.

In Fig. 4, each transaction is labeled (e.g., T1). Performance, recoverability and accessibility are represented by letter 'P', '*R*' and '*A*' respectively. If a quality factor (e.g. accessibility) has more than one metric, we need to represent the name of each metric to avoid confusion; otherwise (e.g., performance and recoverability), we only show its quality value. For example, in the *order reception* QN, '*1s*' means the response time of this transaction is one second; '*10min*' means the recovery time of this transaction is ten minutes; 'data protection (90%)' and 'access control (90%)' means the satisfaction percentages of the two accessibility metrics are both 90%. A "*Null*" value of a quality metric in a certain QN means the metric is irrelevant to this QN. Due to the irrelevance, the satisfaction percentage for a Null value is therefore 100%.

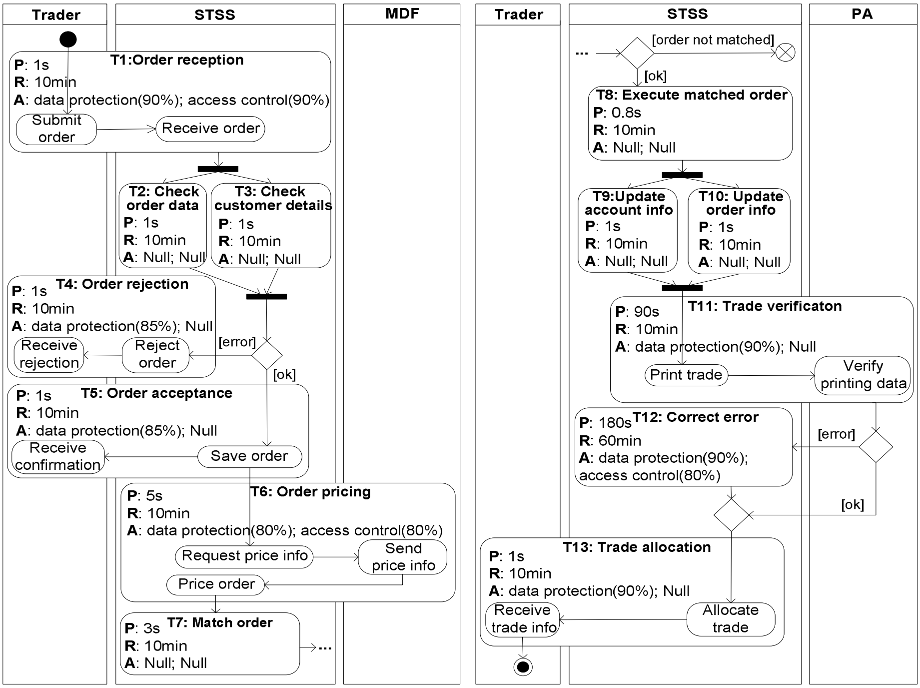

Fig. 4. The QProcess for STSS

3.3 QRs Elaboration

According to the *QRA* meta-process, the QRs elaboration starts from the iterative decomposition of a QP into sub-QPs until each of which matches one of the five basic process flow patterns. For each matched sub-QP, we apply corresponding formulae to calculate the quality metrics of its P, R and A. The interested readers are referred to [4] for a detailed description of the QRs elaboration for STSS.

4 Discussion and Conclusion

The QRA approach is a significant contribution to the requirements engineering field. In this field, there have been a number of approaches available to support QRs. The NFR Framework [12] is the most well known approach. It treats QRs as softgoals and assists their elicitation, specification and evaluation based on a so-called Softgoal Interdependency Graph (SIG). SIGs treat QRs separetely from FRs, whereas in the QRA approach, QRs are treated alongside FRs during goal modeling and business process modeling. In order to integrate the NFR Framework with FRs, Chung and Supakkul [13] attempted to combine SIG with functional use cases. The QRA approach incorporates QRs into functional business processes.

With respect to the incorporation of QRs into business processes, attempts have been made to represent QRs in different process languages [14, 15]. Specifically, Heidari et al. [16] developed a language-independent business process metamodel enriched with different quality factors and metrics. While these approaches focus on the representation of QRs in business processes, they provide little support for further elaboration of QRs. Aburub et al [17] presented an approach to the identification and inclusion of QRs in business processes. Saeedi et al [18] put forward an approach to measure performance, reliability and cost in BPMN..

Compared with the above appraoches, the QRA approach provides a more systematic way to representing, elaborating and measuring QRs alongside FRs. Its contribution is three-fold:

The QRA approach enables a systematic representation of QRs and FRs by providing an integrated modeling language. This language, supported by the RE-Tools [11], combines goal model, functional business process model and QR model into one coherent model. The benefit of such a representation is that business analysts can have a consolidated picture of all user requirements, rather than just fragmented requirements. The representational language of QRA is therefore an effective communication tool to support the early requirements elicitation and discussion.

The QRA approach facilitates a systematic elaboration of QRs and FRs through three levels of goal modeling, business process and QR modeling. Goal modeling allows business analysts to work with customers in a step-wise refinement fashion by elaborating fuzzy, business goals gradually into concrete, operationalized goals. These concrete goals can then be “designed” into business processes as specific business activities. Finally, QR modeling completes business processes by associating each business activity with a set of related QRs. Such a process serves as a basis for communication and discussion between business analysts and customers as well as a basis for communication between business analysts and software architects.

The QRA approach provides a set of formulae for calculating the values of QRs so that QRs can be measured or quantified. These formulae are based on the five control flow patterns of business process modeling and should be applicable to any business process model. Space limitation prevents us from discussing these formulae in detail.

References

1. Ameller, D., Ayala, C., Cabot, J., Franch, X.: Non-functional Requirements in Architectural Decision Making. IEEE Software (2013)
2. Loucopoulos, P., Sun, J., Zhao, L., Heidari, F.: A Systematic Classification and Analysis of NFRs. In: Americas Conference on Information Systems (accepted, 2013)
3. Heesch, U.V., Avgeriou, P.: Mature Architecting—a Survey about the Reasoning Process of Professional Architects. In: IEEE/IFIP Conf. Software Architecture (WICSA 2011), pp. 260–269. IEEE Press, Piscataway (2011)
4. Sun, J., Zhao, L., Loucopoulos, P., Zhou, B.: QRA: A Quality Requirements Analysis Approach for Service Systems. In: International Conference on Services Computing. IEEE Press (2013)
5. Donzelli, P.: A goal driven and agent-based requirements engineering framework. Requirements Engineering Journal 9, 16–39 (2004)
6. Bubenko, J., Rolland, C., Loucopoulos, P., DeAntonellis, V.: Facilitating "fuzzy to formal" requirements modelling. In: IEEE International Conference on Requirements Engineering, pp. 154–157. IEEE Press, Los Alamitos (1994)
7. Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., Robinson, W.: The Brave New World of Design Requirements. Information Systems 36, 992–1008 (2011)
8. Aalst, W.M.P., Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distrib. Parallel Dat. 14, 5–51 (2003)
9. Lamsweerde, A.: Goal-oriented Requirements Engineering: A Guided Tour. In: IEEE International Symposium on Requirements Engineering, pp. 249–262. IEEE Press (2001)
10. Yu, E.S.K.: Towards modelling and reasoning support for early-phase requirements engineering. In: IEEE International Symposium on Requirements Engineering, pp. 226–235. IEEE Press, Los Alamitos (1997)
11. RE-Tools, http://www.utdallas.edu/~supakkul/tools/RE-Tools/
12. Mylopoulos, J., Chung, L., Nixon, B.: Representing and Using Nonfunctional Requirements: A Process-oriented Approach. IEEE T. Software Eng. 18, 483–497 (1992)
13. Chung, L., Supakkul, S.: Representing NFRs and FRs: A Goal-oriented and Use Case Driven Approach. In: Dosch, W., Lee, R.Y., Wu, C. (eds.) SERA 2004. LNCS, vol. 3647, pp. 29–41. Springer, Heidelberg (2006)
14. Korherr, B., List, B.: Extending the EPC with Performance Measures. In: ACM Symposium on Applied Computing, pp. 1265–1266. ACM Press, New York (2007)
15. Pavlovski, C.J., Zou, J.: Non-functional Requirements in Business Process Modeling. In: Asia-Pacific Conference on Conceptual Modelling, pp. 103–112. Australian Computer Society, Darlinghurst (2008)
16. Heidari, F., Loucopoulos, P., Kedad, Z.: A Quality-oriented Business Process Meta-model. In: Barjis, J., Eldabi, T., Gupta, A. (eds.) EOMAS 2011. LNBIP, vol. 88, pp. 85–99. Springer, Heidelberg (2011)
17. Aburub, F., Odeh, M., Beeson, I.: Modelling non-functional requirements of business processes. Inform. Software Tech. 49, 1162–1171 (2007)
18. Saeedi, K., Zhao, L., Sampaio, P.R.F.: Extending Bpmn for Supporting Customer-Facing Service Quality Requirements. In: IEEE International Conference on Web Services, pp. 616–623. IEEE Press, Piscataway (2010)