A Systematic Classification and Analysis of NFRs

Pericles Loucopoulos
Harokopio University of Athens
p.loucopoulos@hua.gr

Jie Sun
Zhejiang University
jianbin@zju.edu.cn

Liping Zhao
The University of Manchester
liping.zhao1@gmail.com

Farideh Heidari
Delft University of Technology
f.heidari@tudelft.nl

ABSTRACT

The main agenda of Requirements Engineering (RE) is the development of tools, techniques and languages for the elicitation, specification, negotiation, and validation of software requirements. However, this development has traditionally been focused on functional requirements (FRs), rather than non-functional requirements (NFRs). Consequently, NFR approaches developed over the years have been fragmental and there is a lack of clear understanding of the positions of these approaches in the RE process. This paper provides a systematic classification and analysis of 89 NFR approaches.

Keywords

Non-functional requirements; NFRs; NFR approaches; quality requirements; requirements engineering; systematic classification and analysis of NFRs approaches.

INTRODUCTION

The importance of Requirements Engineering (RE) has long been established both in research and in practice. There is a significant intellectual activity in the RE field as evidenced by the large volume of papers published in journals and conferences *c.f.* [1] and by the adoption of a variety of techniques by industry *c.f.* [2]. A number of initiatives have been put forward to support the RE lifecycle, initiatives such as risk-driven methodologies [3], requirements tracing [4, 5], model re-use [6], use of scenarios [7], use of visualization [8], use of business rules [9, 10], enterprise modeling [11] and goal modeling [12, 13] to name but a few.

Historically, there has been a distinction between two classes of requirement, functional requirements (FRs) and non-functional requirements (NFRs), the former referring to the operational properties and the latter to the quality properties that the desired system must posses. It is generally agreed that more emphasis has been placed to date on FRs rather than NFRs. However, the changing landscape of requirements demands an increased attention to those aspects of requirements that deal with quality factors, the NFRs. There is a whole set of new advanced design requirements [14] that demand now more than ever close attention to the effectiveness of the SE process. The field's focus and scope have shifted from engineering of individual systems and components towards the generation, adaptation and maintenance of software-intensive ecosystems consisting of software, hardware, human and organizational agents, business processes and more. Such software ecosystems require attention in three areas namely those of design and evolution, orchestration and control, and monitoring and assessment [15].

The context and motivation for this paper is the changing landscape of RE that demands now more than ever particular attention to NFRs. Whilst there is ample evidence (*c.f.* [16]) that NFRs play a significant role in systems, there is surprisingly an absence of an agreed position regarding the definition of NFRs, their classification and presentation [17].

To address this problem, the paper presents a systematic literature review (SLR) of the field of NFRs. Recent years have witnessed an increase in the use of evidence based approaches in SE [18, 19] with a SLR *c.f.* [20] being regarded as the most reliable method. To this end procedures for evidence-based SE are presented in [21, 22]. A study of 33 unique studies, carried out between 2004-2007, revealed 35 SLRs in the field of SE [23]. This paper provides an aggregated view about research efforts to date on NFRs.

The paper is organized as follows. First, the methodology adopted for the classification framework is defined, introducing also the classification scheme. Second, using this classification scheme the 89 approaches are defined according to their relevance in each one of the possible classes of evaluation. Third, an analysis is provided in terms of the contribution of the

89 approaches. Finally, the paper concludes with some reflections and a number of observations for future research challenges in the field of NFRs.

METHODOLOGY FOR THE CLASSIFICATION SCHEME

Kitchenham et al [23] identify two types of SLR, *conventional* SLRs, relating to a specific research question and *mapping* SLRs, concerned with identifying and classifying primary studies in an area. This paper falls in the mapping category. Publications related to NFRs were identified by (a) searching *databases* and *citation indices* (DBLP, Google Scholar), (b) searching *journals* (most prominently, Requirements Engineering journal, IEEE Software, IEEE Transactions on Software Engineering) and (c) searching the proceedings of relevant *conferences* (most prominently, IEEE International RE Conference, International Conference on SE, International Conference on Conceptual Modeling, AMCIS, ICIS). This search revealed 89 candidate approaches.

In order to analyze these approaches a classification scheme was devised that is based on three fundamental perspectives within RE [24]: (a) the *contextual* dimension; (b) the *process* dimension; and (c) the *application* dimension. The choice of these three dimensions as a framework for reviewing NFRS is grounded on the universally accepted methodology of RE in terms of its activities, augmented by two considerations that consider the focus and application of NFRs.

The *contextual* dimension addresses a dichotomy that characterizes requirements as those of early and late requirements. Early requirements [25-27] are almost exclusively driven by client stakeholders' communication. Issues of early requirements include: (a) the customer profiles of a business process, (b) the likely demand for product or service made by each type of customer, (c) the level of desirable service that the business process should strive to achieve, (d) the resources that are required in order to achieve these levels of service and (e) the trade-off between levels of service and requisite resource between all client stakeholders of a business process. Late requirements focus on the attributes of the desired system in terms of both functionality and quality.

The *process* dimension addresses the issue of where a particular approach fits within the RE lifecycle. Within the RE literature, multiple frameworks have been developed, positing anywhere from two to seven primary requirements tasks [28]. For the current review, we adopt a widely-employed and straightforward categorization of the requirements processes into four facets: (i) discovery (more commonly referred to as elicitation), (ii) specification, (iii) negotiation and (iv) validation & verification [29]. During requirements *discovery,* designers develop an understanding of the application domain and infer specific design needs through consultation with stakeholders and other sources [30]. Requirements *specification* is both a noun and a verb in RE parlance. As a noun, a specification is the document in which the requirements for a design effort are articulated thus representing an agreement between the stakeholders and the design team [31, 32]. The verb suggests the process of developing and managing the specification document; it is the process by which the design team acquires, abstracts, and represents the requirements [29, 33]. During requirements *negotiation* stakeholders focus on alternative solutions and attempt to envision potential behavior of the system prior to its implementation [34]. Finally, during *requirements validation and verification* designers ensure that the requirements are (i) of high quality, (ii) by addressing the users' needs, (iii) being appropriate for the design effort, and (iv) having no inconsistencies or defects [35].

The *application* dimension is used to classify approaches to those that focus on either one or more specific NFRs (e.g. security, privacy and performance) or on approaches that are of generic nature that may be used for any kind of NFR.

The 89 approaches reported in this paper were investigated against these three dimensions which when considered in their totality give rise to 16 classification criteria.

CLASSIFICATION OF NFR APPROACHES

This section provides a classification of known NFR approaches, using the multi-dimensional framework described in the preceding section. The results are shown in tabular format in this section. For reasons of readability the overall table is presented in four parts according to the four phases of the process dimension. For each process phase the investigated approaches are classified according to whether they address early or late requirements and for each one of there is a further classification based on the application whether it is aimed at a specific quality characteristic or the approach is generic and therefore potentially applicable to any application. Each entry is given a name and at least one representative reference. Some entries are named according to the name given by the authors of the approach whereas others are introduced in this paper so that overall there is consistency and also an easy way of distinguishing one from another beyond the actual reference.

Table 1 gives a summary of the investigated approaches for the *elicitation* of NFRs. There are altogether 21 approaches with greater emphasis on generic applicability.

Most approaches that deal with early requirements (i.e. examining NFRs from an organizational prism) are goal-oriented, and they usually treat NFRs as soft-goals.

For example, the i* framework [36], founded on the concepts of goal and agent, is capable of capturing organizational knowledge of soft-goals through the identification of organizational agents and their dependencies. The NFR Framework [37-39] elicits soft-goals through a Soft-goal Interdependency Graph. The EKD approach [40] proposes the enterprise goal view, and based on this view, uses a network of goals to capture the causal structure of an enterprise.

Early Requirements		**Late Requirements**	
Specific	***Generic***	***Specific***	***Generic***
• Security Misuse Cases [41], [42] • Usability Soft goal Catalogue [43].	• i* Framework [36] • NFR Framework [37], [38], [39] • EKD [40] • Quality-Based RE Framework [44] • Activity-Based Quality Model [45] • NFR Enhanced Problem Frames [46] • Problem Frames for Soft goals [47] • Layered NFRs Analysis [48].	• Use Cases for Efficiency NFR [49].	• NFR Framework [37], [38], [39] • Quality-Based RE Framework [44] • Activity-Based Quality Model [45] • NFR Classifications [50] • NFR Patterns [51],[52] • NFR Modeling [53] • NFR Aspects [54] • NFR Elicitation from Business Processes [55] • Automatic Text Analysis of NFRs [56, 57] • Experience-Based NFR Decomposition [58]

Table 1. Elicitation Process

For late requirements, there has been much research activity since historically this was the area that first attracted attention by researchers concerned with software quality. In general these approaches fall into three categories: (i) stakeholder engagement [50], [58], (ii) use of patterns [51, 52] and models [53-55], and (iii) identification of NFRs from textual descriptions [56, 57].

Table 2 gives a summary of the investigated approaches for the *specification* of NFRs. There are altogether 30 approaches with greater emphasis on late requirements.

Early requirements approaches for NFRs are mostly generic techniques the majority of which are based on some variation of the goal-oriented paradigm. The NFR Framework [37-39] and i* framework [36] are the most popular specification approaches for generic NFRs. The former uses Soft-goal Interdependency Graphs to specify decompositions and dependencies of NFRs; whereas the latter provides a strategic dependency model and a strategic rationale model that specify early NFRs graphically. Based on the soft-goal concept, there are also textual specification language, such as the informal requirements specification table used by Yin and Jin [46] to represent characteristics and conditions of NFRs, the semi-formal textual specification proposed by Jureta et al. [59] to characterize NFRs, and Formal Tropos [60] that complements graphical i* models with a formal notation.

With regards to late requirements, there have been a number of approaches that focus on specific applications, primarily those of (a) security *c.f.* the works on security use/misuse cases [61, 62], security activity diagrams [63], and security patterns [64]; (b) performance *c.f.* the work on UML-centric approaches [65], semantic frameworks [66], and agile software performance specification [67]; (c) efficiency, by integrating it into use cases *c.f.* [49].

A large number of works that specify generic NFRs have been proposed in the literature. Late NFRs are specified and represented based on UML models [68-73], goal models [51, 74-77], use cases [51, 77, 78], and process models [79-81].

There are also other languages and templates for NFR specification, such as those put forward by [82], quality attributes taxonomy [83] and the Volere Specification Template [84].

Table 3 gives a summary of the investigated approaches focusing on the *negotiation* of NFRs. There are altogether 16 approaches that deal with negotiation issues, the majority of which focus on late requirements with the exception of the use of matrix maps [78], goals [85, 86] and conflict resolution [87].

For the negotiation of specific NFRs, there have been several approaches. To support the identification and resolution between different NFRs, a set of approaches and tools have been proposed. These include the cost-quality tradeoff approach

[88], the Quality Attribute Risk and Conflict Consultant tool [89] and the Software Cost Option Strategy Tool [90]. For trustworthy software, Zhu et al [91] developed a fuzzy qualitative and quantitative soft-goal interdependency graphs approach (FQQSIG). To manage conflicts between security and usability, Mairiza and Zowghi [92] proposed an ontological framework.

Early Requirements		**Late Requirements**	
Specific	***Generic***	***Specific***	***Generic***
• Security Misuse Cases[41],[42]	• NFR Framework [37],[38],[39] • i* Framework [36] • NFR Enhanced Problem Frames [46] • Quality Requirements Analysis [59] • Formal Tropos [60]	• Security Use Cases [61] • Security Misuse Cases[62] • Security Activity Diagram [63] • Security Patterns [64] • UML Based Performance Engineering [65] • Semantic Framework for Performance [66] • Agile Software Performance Spec [67] • Use Cases for Efficiency NFR [49]	• NFR Framework [37],[38],[39] • Quality Requirements Analysis [59] • UML QoS Profile [68] • UML Profile for NFRs [69] • UML Models for NFRs [70],[71],[72] • Integrating NFRs with Object Diagrams [73] • KAOS [74] • Softgoal Composition [75] • NFR Hardgoal Representation[76] • Goals and Use Cases for NFRs [51],[77] • Use Cases for NFRs [78] • ProcessNFL [79] • Quality-Enhanced BPMN [81],[80] • NOn-FUNctional Language [82] • Quality Attributes Taxonomy [83] • Volere [84]

Table 2. Specification Process

Early Requirements		**Late Requirements**	
Specific	***Generic***	***Specific***	***Generic***
	• Matrix Maps [85] • GBRAM [86] • NFR Conflict Resolution [87]	• Cost-Quality Tradeoff [88] • Quality Requirements Conflict Identification [89] • Software Cost Option Strategy Tool [90] • FQQSIG [91] • Ontological Framework for Security & Usability [92]	• Groupware for Requirements Negotiation [93],[94] • A Catalog of Conflicts [95] • Ontology-Based Conflict Analysis [96] • Extended NFR Framework [76], [97] • NFR Optimization [98] • NFR Traceability [99], [100] • Hierarchical Constraint Logic Programming [101] • Personal Construct Theory [102]

Table 3. Negotiation Process

To manage conflicts among late NFRs, there are a variety of generic approaches including conflict identification, analysis and resolution. The most prominent approach is the Win-Win groupware system [93, 94]. It determines both FRs and NFRs as negotiated win conditions, based on the WinWin Spiral Model. There are other approaches also such as those using conflict catalogues [95] and ontology [96]. The NFR Framework, already presented in the elicitation and specification tables has been widely extended [76, 97] and adapted [98] in different ways to identify, analyze and resolve various NFR conflicts. Modeling requirements traceability is another method to address NFR conflicts [99, 100]. Other theories include Hierarchical Constraint Logic Programming [101] and Personal Construct Theory [102].

Table 4 gives a summary of the investigated approaches focusing on the *validation and verification* of NFRs. There are altogether 22 approaches that address issues of validation and verification.

For early requirements there are approaches that deal with specific applications, specifically, with reliability [103] using UML models and with performance by adapting the NFR Framework [104], the later also being of relevance to late requirements. Within early requirements there are many approaches that are of generic nature of which the NFR Framework is used [37-39] to provide a label propagation mechanism to qualitatively evaluate NFR achievement. The label propagation mechanism has been extended to support quantitative evaluation [105-107]. The validation check list [48] and traceability management [108] are also frequently-used to validate NFRs. All these approaches, except [48], can be used for late NFRs as well.

Early Requirements		**Late Requirements**	
Specific	***Generic***	***Specific***	***Generic***
• Bayesian Reliability Prediction [103] • Performance Requirements Framework [104]	• NFR Framework [37],[38],[39] • Quantitative NFR Patterns [105] • Quantitative Assessment of NFRs [106] • Reasoning About Partial Goal Satisfaction [107] • Validation Check List [48] • Traceability Management [108]	• Performance Requirements Framework [104] • Specification Quality Gate [109] • Reliability Prediction [110]	• NFR Framework [37],[38],[39] • Quantitative NFR Patterns [105] • Quantitative Assessment of NFRs [106] • Reasoning About Partial Goal Satisfaction [107] • Traceability Management [108] • UML-Based Verification [111] • Scenario-Based Assessment [112] • Bayesian Belief Network [113] • Contextual Information Ontology [114] • Automata-Based Verification [115] • Abstract Interpretation Based Verification [116]

Table 4 Validation and Verification Process

For late requirements, specific applications have been targeted such as performance [109] and reliability [110] of systems. There are also approaches that have been proposed from a general perspective dealing independently of any application with issues such as verification using UML models [111] or using scenarios for assessing the degree of validity of NFRs [112], or using formal methods, such as Bayesian belief network [113], ontology [114], finite state automata [115], and abstract interpretation [116] for verification purposes.

ANALYSIS OF NFR APPROACHES BASED ON THE CLASSIFICATION SCHEME

Using the above classification, three different analysis were performed: (a) process-oriented analysis to analyze the numeric impact of approaches in each one of the 16 different options, (b) single dimensional analysis to examine the percentage of approaches focused along each one of the three dimensions and (c) multi-aspect analysis within a single dimension to demonstrate the degree to which investigated approaches address than one aspect within a dimension.

Figure 1 demonstrates the process-oriented analysis. This presentation distinguishes a number of approaches in each process within different aspects of the other two dimensions namely, context and application. In other words, the number of approaches covering each combination of <process, application (specific/generic)> and <process, context (early/late) > are provided in figure 1. The analysis is in terms of absolute values.

This analysis reveals that, in all 4 phases generic approaches are much more prominent. This is not surprising since most new research focuses on the provision of theories, models and techniques at an abstract level. However, the paucity of approaches that tackle specific aspects of quality is a cause for concern because ultimately, NFRs must address specific quality requirements whether early or late, requirements such as compliance, security, privacy, reliability, robustness and so on.

The graph of Figure 1 also reveals that there is more emphasis on late requirements rather than early requirements, right across all 4 phases of the NFRs lifecycle. In other words, the emphasis is mostly on requirements for software and systems rather than on organizational and human factors. And yet research and practice have shown that there is an intertwining between requirements for systems and requirements for contexts of systems [14].

Figure 2 summarizes the single-dimension analysis and compliments the findings presented in Figure 1, by presenting a comparative analysis for context, process and application. For each dimension, the percentage of approaches focused on different aspects of the dimension is presented.

For context and application the comparative influences have already been discussed. For the process dimension, the specification phase is the phase that has received most attention and negotiation the least. The bias towards specification is perhaps due to the preference of researchers in the area to focus on the innovation of new modeling techniques rather than any on the other issues. However, the relatively small effort in negotiation, coupled with the lack of effort in early requirements seems to indicate that human-centered and contextual issues do not receive as much attention as they should.

Figure 3 summarizes the analysis of two dimensions namely process and context. The investigated approaches were analyzed to ascertain to what degree an approach covers one or more of the process phases. The analysis also focused on the degrees to which approaches focused on one or the other of the two contexts and also those that consider both contexts.

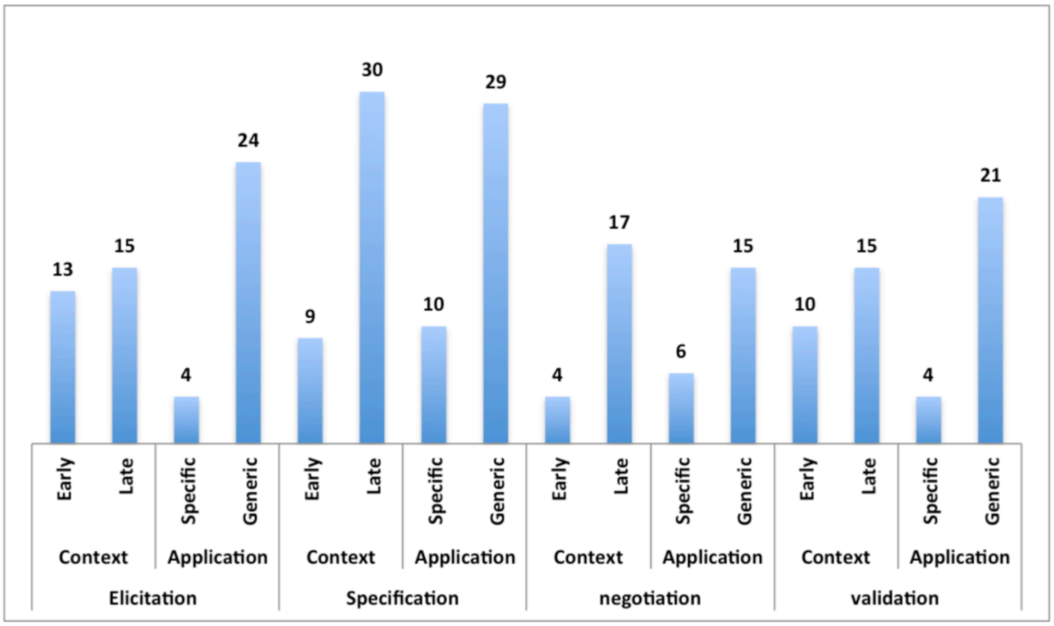

Figure 1. Process-oriented analysis

Figure 2. Analysis against three dimensions

The graphs of Figure 3 reveal the fragmentation of the field in both process and context. In terms of process, the majority of approaches (87%) deal with only one of the well-accepted phases of the requirements lifecycle. A small number of approaches (4%) deal with 3 of the 4 phases. In general, it seems that there is scope for the development of approaches that embrace all lifecycle phases. This is a methodological issue that needs some attention. In terms of context only a small number of approaches (7%) cover both early and late requirements in other words having the ability to seamlessly move from human-centered to system-centered NFRs.

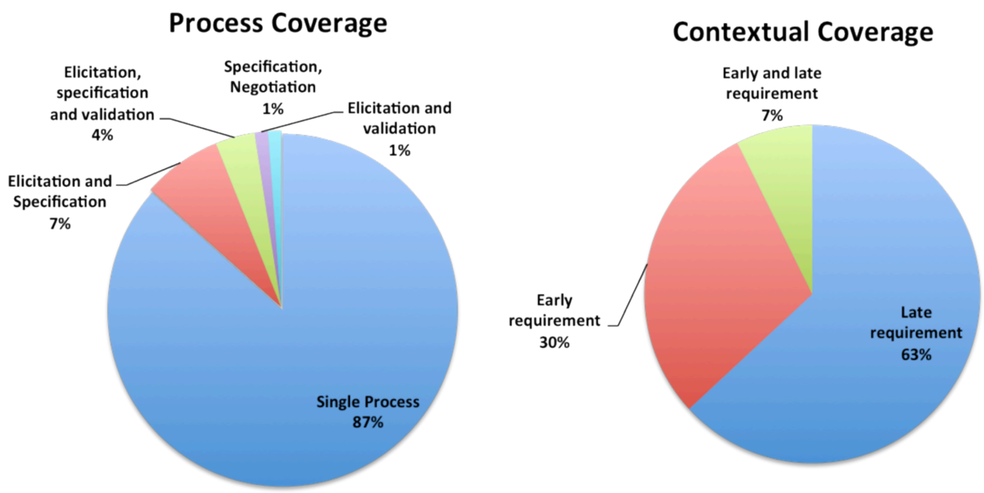

Figure 3. Multi-dimensional analysis

CONCLUSIONS

The main contribution of this paper is the establishment of a clear understanding of the NFR approaches in RE. The paper has presented this understanding through a systematic classification and analysis of 89 NFR approaches. Two important conclusions can be drawn from this analysis:

- *Lack of NFR approaches for early requirements*. One important finding of this paper is that most current NFR approaches are for late rather than early requirements. This seems to suggest that early requirements are more challenging, as they tend to be informal and expressed in natural language. Consequently, they are difficult to systematize. Future research should look into text analysis and natural language processing approaches for the treatment of early NFRs.
- *Lack of domain-specific NFR approaches*. Among 89 NFR approaches we have sampled, only 18 are domain-specific approaches. It seems that the field has not settled yet as to whether generic approaches can address NFRs for all application domains or there is a need for specific NFR approaches for different categories of application (e.g. security, privacy, compliance etc.). Future research needs to focus on the ontology of different applications and through this to ascertain whether generic approaches are adequate for covering all quality aspects of such application domains or there is a need for focusing on specific approaches.

There are a number of implications from this analysis regarding the utility of NFRs with respect to technological advances in the field of information systems, challenges emerging from areas such as service computing, cloud computing and mobile computing.

Indeed, our initial finding [117] shows that most of today's service system development approaches focus mainly on the system facing quality requirements (the *late NFRs*), rather than the customer facing quality requirements (the *early NFRs*). This suggests that despite of the latest advances in the state of the art and practice of software development, the treatment of NFRs remains to be a challenge. According to the authors in [118], this challenge is due to the four characteristics of NFRs:

- NFRs are system-level requirements which cannot be assigned directly to individual system components; instead, they need to be planned at the infrastructure-level as a whole, with their design aspects then entrusted to system components.
- NFRs are often interconnected and their realization in a system requires a collective and coordinated behavior of the system components and a system-level design strategy.
- NFRs are application-specific and their fulfillment necessarily requires a specific design approach appropriate to their applications.
- NFRs are not only a design time concern, but most crucially a runtime concern. Satisfying NFRs in a system means designing runtime mechanisms that can maintain the system's NFRs throughout the execution time.

Based on these characteristics, our future work will also investigate ways to measure or evaluate different NFR approaches.

REFERENCES

Part I: General References

1 Cheng, B.H.C. and Atlee, J.M., Current & future research directions in requirements engineering, in Design requirements engineering: A ten-year perspective, K. Lyytinen, et al., Editors. 2009, Springer.

2 Lyytinen, K.J., Loucopoulos, P., Mylopoulos, J., and Robinson, W., eds. Design requirements engineering: A ten-year perspective. Lecture notes in business information processing. 2009, Springer: Heidelberg.

3 Boehm, B.W. (1988) A spiral model of software development and enhancement. *Computer*. 21, 5 61-72.

4 Gotel, O.C.Z. and Finkelstein, C.W. (1994) An analysis of the requirements traceability problem. *Proceedings of the First International Conference on Requirements Engineering*. Colorado Springs, CO, USA.

5 Ramesh, B. and Jarke, M. (2001) Toward reference models for requirements traceability. *IEEE Transactions on Software Engineering*. 27, 1 58-93.

6 Rolland, C., Loucopoulos, P., Grosz, G., and Nurcan, S. (1998) A framework for generic patterns dedicated to the management of change in the electricity supply industry. *International DEXA Conference (DEXA98)*. Vienna, Austria.

7 Kavakli, E., Loucopoulos, P., and Filippidou, D. (1996) Using scenarios to systematically support goal-directed elaboration for information systems requirements. *IEEE Symposium and Workshop on Engineering of Computer-Based Systems*. Friedrichshafen, Germany: IEEE Computer Society.

8 Lalioti, V. and Loucopoulos, P. (1994) Visualisation of conceptual specifications. *Information Systems*. 19, 3 291-309.

9 Loucopoulos, P. and Katsouli, E. (1992) Modelling business rules in an office environment. *SIGOIS Bulletin*. 13(2), 28-37.

10 Tsalgatidou, A. and Loucopoulos, P. (1990) An object-oriented rule-based approach to the dynamic modelling of information systems. *International Conference on Dynamic Modelling of Information Systems*. Noordwijkerhout, The Netherlands.

11 Loucopoulos, P., From information modelling to enterprise modelling, in Information systems engineering: State of the art and research themes, S. Brinkkemper, E. Lindencrona, and A. Solvberg, Editors. 2000, Springer 67-78.

12 Kavakli, E. and Loucopoulos, P. (2006) Experiences with goal-oriented modelling of organisational change. *IEEE Transactions on Systems, Man and Cybernetics - Part C*. 36, 2 221-235.

13 Yu, E.S.K. and Mylopoulos, J. (1998) Why goal oriented requirements engineering.

14 Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., and Robinson, W. (2011) The brave new world of design requirements. *Information Systems*. 36, 7 992-1008.

15 SEI (2006) Ultra large scale systems: The software challenge of the future.

16 Ameller, D., Ayala, C., Cabot, J., and Franch, X. (2013) Non-functional requirements in architectural decision making. *IEEE Software*, March/April 2013.

17 Glinz, M. (2007) On non-functional requirements. *15th IEEE international requirement engineering conference*. IEEE.

18 Dybå, T., Kitchenham, B.A., and Jørgensen, M. (2005) Evidence-based software engineering for practitioners. *IEEE Software*. 22, 1 58-65.

19 Kitchenham, B.A., Dybå, T., and Jørgensen, M. (2004) Evidence-based software engineering. *International Conference on Software Engineering (ICSE '04)*. Washington DC, USA: IEEE Computer Society.

20 Khalid, K.S., Kunz, R., Kleijnen, J., and Antes, G., Systematic reviews to support evidence-based medicine. 2003: Springer.

21 Kitchenham, B.A. and Charters, S. (2007) Guidelines for performing systematic literature reviews in software engineering. EBSE-2007-01

22 Kitchenham, B.A. (2004) Procedures for undertaking systematic reviews. TR/SE-0401

23 Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O.P., Turner, M., and Mahmood Niazi, S.L. (2010) Systematic literature reviews in software engineering – a tertiary study. *Information and Software Technology*. 2010, 52 792-805.

24 Loucopoulos, P. and Karakostas, V., System requirements engineering. 1st ed. International series in software engineering, ed. D. Ince. 1995, London: McGraw Hill 160.

25 Fuxman, A., Liu, L., Pistore, M., Roveri, M., and Mylopoulos, J. (2003) Specifying and analyzing early requirements: Some experimental results. *Proceedings of the 11th IEEE International Conference on Requirements Engineering*. Monterey Bay, California.

26 Loucopoulos, P. (2004) Engaging stakeholders in defining early requirements. *6th International Conference on Enterprise Information Systems (ICEIS 2004)*. Porto, Portugal.

27 Wegmann, A., Julia, P., Regev, G., Perroud, O., and Rychkova, I. (2007) Early requirements and business-it alignment with seam for business. *15th IEEE International Requirements Engineering Conference*,

28 Thayer, R.H. and Dorfman, M., Software requirements engineering. 2nd ed. 1997, Los Alamitos, Calif.: IEEE Computer Society Press xvii, 483 , [1] folded leaf of plates.

29 Loucopoulos, P. and Karakostas, V., System requirements engineering. Mcgraw-hill international series in software engineering. 1995, London ; New York: McGraw-Hill Book Co x, 160.

30 Sommerville, I. and Sawyer, P., Requirements engineering : A good practice guide. 1997, Chichester, England ; New York: John Wiley & Sons 391.

31 Jackson, M.J., Software requirements & specifications : A lexicon of practice, principles, and prejudices. Acm press books. 1995, New York, Wokingham, England ; Reading, Mass.: ACM Press ; Addison-Wesley Pub. Co. xvi, 228.

32 Ghezzi, C., Jazayeri, M., and Mandrioli, D., Fundamentals of software engineering. 2002: Prentice Hall PTR Upper Saddle River, NJ, USA.

33 Vessey, I. and Conger, S. (1994) Requirements specification: Learning object, process, and data methodologies. *Communications of the ACM*. 37, 5 102-113.

34 Hillier, B., Musgrove, J., and O'Sullivan, P., Knowledge and design, in Developments in design methodology, N. Cross, Editor. 1984, John Wiley & Sons, Inc. 245-264.

35 Boehm, B. (1984) Verifying and validating software requirements and design specifications. *IEEE Software*. 1, 1 75-88.

117 Sun, J., Zhao, L., Loucopoulos, P., and Zhou, B. (2013) Qra: A quality requirements analysis approach for service systems. *Proceedings of International Conference on Services Computing*, Accepted,

118 Wang, X., Sun, J., Yang, X., Wang, Y., Li, S., and Kavs, A.J. (2012) Deriving problem frames from business process and object analysis models. *Expert Systems*. Published online.

Part II: NFRs-approach Specific References

36 Yu, E.S.K. (1997) Towards modelling and reasoning support for early-phase requirements engineering. *IEEE International Symposium on Requirements Engineering*, 226- 235.

37 Mylopoulos, J., Chung, L., and Nixon, B. (1992) Representing and using nonfunctional requirements: A process-oriented approach. *IEEE Transactions on Software Engineering*. 18, 6 483-497.

38 Chund, L. and Nixon, B.A. (1995) Dealing with non-functional requirements three experimental studies of a process-oriented approach. *ICSE95*, 25.

39 Chung, L., Nixon, B.A., Yu, E., and Mylopoulos, J. (2000) Non-functional requirements in software engineering. *International Series in Software Engineering*. 5, 476.
40 Loucopoulos, P. and Kavakli, V. (1999) Enterprise knowledge management and conceptual modelling. *International Conference on Conceptual Modeling*, Springer Press, 123-143
41 Herrmann, A. and Peach, B. (2008) Moqare: Misuse-oriented quality requirements engineering. *Requirements Engineering*. 13, 1 73-86.
42 Alexander, I. (2003) Misuse cases help to elicit non-functional requirements. *Computing&Control Engineering Journal*. 14, 1 40-45.
43 Cysneiros, L.M., Werneck, V.M., and Kushniruk, A. (2005) Reusable knowledge for satisficing usability requirements. *RE05*, 463-464.
44 Donzelli, P. and Bresciani, P. (2004) Improving requirements engineering by quality modelling-a quality-based requirements engineering framework. *Journal of Research and Practice in Information Technology*. 36, 277-294.
45 Wagner, S., Deissenboeck, F., and Winter, S. (2010) Managing quality requirements using activity based quality models. *Proceedings of the 6th International Workshop on Software Quality*, ACM Press, 29-34.
46 Yin, B. and Jin, Z. (2012) Extending the problem frames approach for capturing non-functional requirements. *ICIS12*, 432-437.
47 Supakkul, S. and Chung, L. (2009) Extending problem frames to deal with stakeholder problems: An agent- and goal-oriented approach. *SAC09*, 389-394.
48 Rao, A.A. and Gopichand, M. (2011) Four layered approach to non-functional requirements analysis. *International Journal of Computer Science*. 8, 6 371-379.
49 Dorr, J., Kerkow, D., Knethen, A., and Paech, B. (2003) Eliciting efficiency requirements with use cases. *REFSQ03*, 37-46.
50 Cleland-Huang, J., Settimi, R., Zou, X., and Solc, P. (2007) Automated classification of non-functional requirements. *Requirements Engineering*. 12, 2 103-120.
51 Supakkul, S. and Chung, L. (2005) Integrating frs and nfrs: A use case and goal driven approach. *SERA05*, 30–37.
52 Chung, L. and Supakkul, S. (2006) Capturing and reusing functional and non-functional requirements knowledge: A goal-object pattern approach. *IEEE International Conference on Information Reuse and Integration*, 539-544.
53 Fatwanto, A. and Boughton, C. (2008) Analysis, specification and modeling of non-functional requirements for tanslative model-driven development. *CIS08*, 405-410.
54 Yu, Y., Leite, J., and Mylopoulos, J. (2004) From goals to aspects discovering aspects from requirements goal models. *Proceedings of IEEE International Symposium on Requirements Engineering*, IEEE Press, 38-47
55 Jaramillo, A.F. (2011) Non-functional requirements elicitation from business process models. *RCIS11*, 1-7.
56 Cleland-Huang, J., Settimi, R., Zou, X., and Solc, P. (2006) The detection and classification of non-functional requirements with application to early aspects. *Proceedings of 14th IEEE International Requirements Engineering Conference*, 39-48.
57 Casamayor, A., Godoy, D., and Campo, M. (2010) Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. *Information and Software Technology*. 52, 4 436-445.
58 Doerr, J., Kerkow, D., Koenig, T., Olsson, T., and Suzuki, T. (2005) Non-functional requirements in industry-three case studies adopting an experience-based nfr method. *Proceedings of the 13th IEEE International Confere nce on Requirements Engineering*, IEEE Press, 373-382
59 Jureta, I., Faulkner, F., and Schobbens, P. (2006) A more expressive softgoal conceptualization for quality requirements analysis. *Proceedings of the 25th International Conference on Conceptual Modeling*, Springer Press, 281-295.
60 Castro, J., Kolp, M., and Mylopoulos, J. (2001) A requirements-driven software development methodology. *Proceedings of the 13th International Conference on Advanced Information Systems Engineering*, Springer Press, 108-123.
61 Firesmith, D.G. (2003) Security use cases. *Journal of Object Technology*. 2, 3 53-64.
62 Sindre, G. and Opdahl, A.L. (2005) Eliciting security requirements with misuse cases. *Requirements Engineering*. 10, 1 34-44.
63 Rodriguez, A., Fernandez-Medina, E., and Piattini, M. (2006) Security requirement with a uml 2.0 profile. *Proceedings of the 1st International Conference on Availability, Reliability and Security*, 670-677.
64 Lamsweerde, A. (2004) Elaborating security requirements by construction of intentional anti-models. *Proceedings of the 26th International Conference on Software Engineering* IEEE Press, 148-157.
65 Dimitrov, E., Schmietendorf, A., and Dumke, R. (2002) Uml-based performance engineering possibilities and techniques. *IEEE Software*. 19, 1 74-83.
66 Zschaler, S. (2004) Towards a semantic framework for non-functional specifications of component-based systems. *Proceedings of the 30th EUROMICRO Conference*, 92-99.
67 Ho, C., Johnson, M.J., Williams, L., and Maximilien, E.M. (2006) On agile performance requirements specification and testing. *Proceedings of the conference on AGILE*, 47-52.
68 OMG (2009) Umltm profile for modeling quality of service and fault tolerance characteristics and mechanisms specification. formal/2009-04-05
69 Zhu, L. and Gorton, I. (2007) Uml profiles for design decisions and non-functional requirements. *SHARK-ADI07*, 8.
70 Cysneiros, L., Leite, J., and Neto, J. (2001) A framework for integrating non-functional requirements into conceptual models. *Requirements Engineering*. 6, 2 97-115.
71 Cysneiros, L.M. and Leite, J. (2002) Non-functional requirements from elicitation to modelling languages. *Proceedings of the 24th International Conference on Software Engineering*, IEEE Press, 699-700.
72 Cysneiros, L.M. and Leite, J. (2001) Using uml to reflect non-functional requirements. *Proceedings of the 11th Conference of the Centre For Advanced Studies on Collaborative Research*, 2.
73 Tonu, S.A. and Tahvildari, L. (2005) Towards a framework to incorporate nfrs into uml models. *RETR05*, 13-18.
74 Lamsweerde, A. (2001) Goal-oriented requirements engineering: A guided tour. *RE01*, 249-262.
75 Wei, B., Jin, Z., and Liu, L. (2010) A formalism for extending the nfr framework to support the composition of the goal trees. *Proceedings of the 17th Asia Pacific Software Engineering Conference*, 23-32.
76 Kassab, M., Daneva, M., and Ormandjieva, O. (2007) Scope management of non-functional requirements. *EUROMICRO07*, 409-417.
77 Chung, L. and Supakku, S. (2005) Representing nfrs and frs: A goal-oriented and use case driven approach. *SERA04*, 29-41.
78 Cysneiros, L.M. and Leite, J. (2001) Driving non-functional requirements to use cases and scenarios. *XV Brazilian Symposium on Software Engineering*, 7-20.
79 Rosa, N.S., Paulo, P.R.F., and Justo, G.R.R. (2002) Process nfl: A language for describing non-functional properties. *HICSS02*, 3676-3685.
80 Pavlovski, C.J. and Zou, J. (2008) Non-functional requirements in business process modeling. *APCCM08*, 103-112.
81 Sindhgatta, R. and Thonse, S. (2005) Functional and non-functional requirements specification for enterprise applications. *PROFES05*, 189-201.
82 Botella, P., Burgues, X., Franch, X., Huerta, M., and Salazar, G. (2001) Modelling non-functional requirements. *Proceedings of Jornadas Ingenierade Requisitos Aplicados*, 4.
83 Barbacci, M., Klein, M.H., Longstaff, T.A., and Weinstock, C.B. (1995) Quality attributes. CMU/SEI-95-TR-021
84 Robertson, J. and Robertson, S. (2000) Volere: Requirements specification template,
85 Abdul, H., Jamil, A., and Imran, U. (2010) Conflicts identification among non-functional requirements using matrix maps. *World Academy of Science Engineering and Technology*. 44, 1004-1009.
86 Anton, A.I., Bolchini, D., and He, Q. (2003) The use of goals to extract privacy and security requirements from policy statements. TR-2003-17
87 Sadana, V. and Liu, X.F. (2007) Analysis of conflicts among non-functional requirements using integrated analysis of functional and non-functional requirements. *Proceedings of the 31st Annual International Conference on Computer Software and Applications* IEEE Press, 215-218

88 Egyed, A., Biffl, S., Heindl, M., and Grünbacher, P. (2005) Determining the cost-quality trade-off for automated software traceability. *Proceedings of the 20th IEEE/ACM international Conference on Automated Software Engineering*, 360-363.
89 Boehm, B. and In, H. (1996) Identifying quality-requirement conflicts. *IEEE Software*. 13, 2 25-35.
90 Boehm, B. and In, H. (1996) Software cost option strategy tool (s-cost). *Proceedings of the 20th International Conference on Computer Software and Applications*, 15-20.
91 Zhu, M., Luo, X., Chen, X., and Wu, D. (2012) A non-functional requirements tradeoff model in trustworthy software. *Information Sciences*. 191, 61-75.
92 Mairiza, D. and Zowghi, D. (2010) An ontological framework to manage the relative conflicts between security and usability requirements. *Proceedings of the 3rd International Workshop on Managing Requirements Knowledge*, IEEE Press, 1-6
93 In, H., Boehm, B., Rodgers, T., and Deutsch, M. (2001) Applying winwin to quality requirements: A case study *Proceedings of the 23rd International Conference on Software Engineering*, 555-564
94 Boehm, B., Grunbacher, P., and Briggs, R.O. (2001) Developing groupware for requirements negotiation: Lessons learned. *IEEE Software*. 18, 3 46-55.
95 Mairiza, D. and Zowghi, D. (2010) Constructing a catalogue of conflicts among non-functional requirements. *ENASE10*, 31-44.
96 Liu, C. (2010) Ontology-based conflict analysis method in non-functional requirements. *Proceedings of the IEEE/ACIS International Conference on Computer and Information Science*, IEEE Press, 491-496.
97 Hill, R. and Wang, J. (2004) Quantifying non-functional requirements: A process oriented approach. *Proceedings of the 12th International Conference on Requirement Engineering*, IEEE Press, 352-353
98 Burgess, C., Krishna, A., and Jiang, L. (2009) Towards optimising non-functional requirements. *Proceedings of the 9th International Conference on Quality Software*, 269-277.
99 Chung, L., Nixon, B.A., and Yu, E. (1995) Using non-functional requirements to systematically support change. *Proceedings of 2nd International Symposium on Requirements Engineering*, 132-139.
100 Egyed, A. and Grunbacher, P. (2004) Identifying requirements conflicts and cooperation: How quality attributes and automated traceability can help. *IEEE Software*. 21, 6 50-58.
101 Guan, Y. and Ghose, A.K. (2005) Use constraint hierarchy for non-functional requirements analysis. *Proceedings of the 5th International Conference on Web Engineering*, Springer Press, 104-109.
102 González-Baixauli, B., Leite, J., and Laguna, M.A. (2006) Eliciting non-functional requirements interactions using the personal construct theory. *Proceedings of the 14th IEEE International Requirements Engineering Conference*, IEEE Press, 347-348.
103 Cortellessa, V., Singh, H., and Cukic, B. (2002) Early reliability assessment of uml based software models. *Proceedings of the 3rd international workshop on Software and performance*, 302-309.
104 Nixon, B.A. (2000) Management of performance requirements for information systems. *IEEE Transactions on Software Engineering*. 26, 12 1122-1146.
105 Supakkul, S., Hill, T., Chung, L., and Leite, T. (2010) An nfr pattern approach to dealing with nfrs. *RE10*, 179-188.
106 Kassab, M., Daneva, M., and Ormandjieva, O. (2007) Early quantitative assessment of non-functional requirements.
107 Letier, E. and Lamsweerde, A. (2004) Reasoning about partial goal satisfaction for requirements and design engineering. *FSE04*, 53-62.
108 Cleland-Huang, J., Settimi, R., BenKhadra, O., Berezhanskaya, E., and Christina, S. (2005) Goal-centric traceability for managing non-functional requirements. *Proceedings of the 27th International Conference on Software Engineering*, IEEE Press, 362-371.
109 Salger, F., Engels, G., and Hofmann, A. (2009) Inspection effectiveness for different quality attributes of software requirement specifications: An industrial case study. *Proceedings of the 7th ICSE Workshop on Software Quality*, 15-21.
110 Rodrigues, G., Rosenblum, D., and Uchitel, S. (2005) Using scenarios to predict the reliability of concurrent component-based software systems. *Proceedings of the 8th International Conference on Fundamental Approaches to Software Engineering*, 111-126.
111 Ghezzi, C. and Sharifloo, A.M. (2013) Model-based verification of quantitative non-functional properties for software product lines. *Information and Software Technology*. 55, 3 508-524.
112 Gregoriades, A. and Sutcliffe, A. (2007) Scenario-based assessment of nonfunctional requirements. *IEEE Transactions on Software Engineering*. 31, 5 392-409.
113 Zhang, H., Jarzabek, S., and Yang, B. (2003) Quality prediction and assessment for product lines. *Proceedings of the 15th International Conference on Advanced Information Systems Engineering*, Springer Press, 681-695.
114 Molina, H. and Olsina, L. (2007) Towards the support of contextual information to a measurement and evaluation framework. *Proceedings of the 6th International Conference on the Quality Information and Communications Technology*, 154-166.
115 Sun, H., Basu, S., Lutz, R., and Honavar, V. (2009) Automata-based verification of non-functional requirements in web service composition,
116 Cortesi, A. and Logozzo, F. (2005) Abstract interpretation-based verification of non-functional requirements. *Proceedings of the International Conference on Coordination Models and Languages* Springer Press, 49-62.