

# How to specify Non-functional Requirements to support seamless modeling?

## A Study Design and Preliminary Results

Jonas Eckhardt, Daniel Méndez Fernández, Andreas Vogelsang

Technische Universität München

Garching b. München, Germany

{eckharjo,mendezfe,vogelsan}@in.tum.de

**Abstract—Context:** Seamless model-based development provides integrated chains of models, covering all software engineering phases. Non-functional requirements (NFRs), like reusability, further play a vital role in software and systems engineering, but are often neglected in research and practice. It is still unclear how to integrate NFRs in a seamless model-based development. **Goal:** Our long-term goal is to develop a theory on the specification of NFRs such that they can be integrated in seamless model-based development. **Method:** Our overall study design includes a multi-staged procedure to infer an empirically founded theory on specifying NFRs to support seamless modeling. In this short paper, we present the study design and provide a discussion of (i) preliminary results obtained from a sample, and (ii) current issues related to the design. **Results:** Our study already shows significant fields of improvement, e.g., the low agreement during the classification. However, the results indicate to interesting points; for example, many of commonly used NFR classes concern system modeling concepts in a way that shows how blurry the borders between functional and NFRs are. **Conclusions:** We conclude so far that our overall study design seems suitable to obtain the envisioned theory in the long run, but we could also show current issues that are worth discussing within the empirical software engineering community.

The main goal of this contribution is not to present and discuss current results only, but to foster discussions on the issues related to the integration of NFRs in seamless modeling in general and, in particular, discussions on open methodological issues.

**Keywords—Requirements, Requirements Engineering, Non-functional Requirements, Seamless Modeling**

### I. INTRODUCTION

The increasing complexity in software and system development projects results in a demand for expressiveness, modularity, reusability, and analyzability of modeling and specification approaches. Model-based development is the key to meet this demand as it allows to abstract from implementation details and to increase the overall level of abstraction. Yet, model-based development alone does not solve anything as various models still have to be integrated into a holistic chain. In response to this, the idea of seamless model-based development emerged [1]. Seamless modeling aims at elaborating integrated chains of models covering all phases from requirements engineering to system design and verification. The central ingredient of seamless model-based development is a system model that provides the theoretical framework interconnecting all models.

Non-functional requirements (NFRs) further play a vital role in software and systems engineering. There is much work

available in the field of NFRs characterizing single classes of NFRs and classifications as well, such as security and reusability. Yet, we are still far from having a common understanding on the notion of NFRs, let alone from having commonly accepted and integrated taxonomies for NFRs [2], [3], [4] that go beyond a rather abstract level, or even an integration of NFRs in a common system model. In fact, the integration of NFRs and model-based development forms a high priority scope of current research projects that aim at better understanding how practitioners integrate NFRs in context of model-based development and, in particular, what problems they experience [5], [6].

Therefore, it currently remains unclear how to integrate NFRs in seamless model-based development, as the integration in a common system model is not in scope of available contributions. This forms the objective of our ongoing research. In the long run, we want to provide an approach for specifying NFRs such that they can be integrated in a common system model, and thus, supporting seamless model-based development that also takes into account the specification of NFRs.

To reach this goal, we designed an overall study that starts with analyzing how practitioners specify NFRs. This literature-agnostic view allows for getting an overview of the information and structure necessary to specify NFRs sufficiently suitable for subsequent development activities (even if not integrated). Another reason why we base our work on practical data is that we want our resulting theory to emphasize the practical impact rather than the theoretical one alone. Having understood the basic constructs used to specify NFRs in practice, we analyze in a second step the relation between classes of NFRs and the various system modeling dimensions. We use this classification to elaborate, in a third step, a theory on specifying NFRs in context of seamless development, before eventually disseminating and evaluating our theory again in practical contexts.

In this paper, we present our overall design and discuss current results and methodological issues arising from the preliminary analysis of a sample. Please note that the primary aim of this paper is not to present the results alone (as the study design still might be subject to change), but to foster discussions and exchange ideas on this difficult area characterized by various (empirical) challenges.

### II. OVERALL STUDY DESIGN

The goal of the overall study is to analyze natural language NFRs taken from industrial requirements specifications in order

to understand how classes of NFRs relate to existing system modeling dimensions. This serves as a basis for developing a theory for the specification of NFRs to support seamless modeling.

### A. Research Questions

To reach our goal, we formulate the following research questions (RQs):

**RQ1:** *What classes of NFRs are documented in practice and what is their scope.* This RQ examines the state of practice, i.e., what classes of NFRs are actually documented and whether they refer to the context, to the system or to a sub-system.

**RQ2:** *How do classes of NFRs relate to existing system modeling dimensions?* In this RQ, we lay the foundation of the later theory building: we analyze how (and if) classes of NFRs relate to specific system modeling dimensions.

**RQ3:** *How can NFRs be specified to support seamless modeling?* For those NFRs that are related to system modeling dimensions, we build in a third step a theory on how to specify NFRs to support seamless modeling.

**RQ4:** *What are the limitations of specifying NFRs in a seamless context?* In a last step, we analyze the limitations of the resulting theory which we also plan to use for further adjustments of the concepts captured in the theory.

Figure 1 depicts an overview of the overall study design. First, we perform a preliminary classification of a sample (approx. 5%) to see how the individual classifications align, followed by a discussion with an agreement. Based on this discussion, we build a decision tree<sup>1</sup> for the classification to make the classification more transparent. Using this decision tree, we validate the classification on another random sample (approx. 5%), followed again by a discussion with an agreement. Finally, to answer RQ1 and RQ2, we analyze the whole data set (not in scope of the paper at hands). While RQ1 and RQ2 are concerned with the collection and classification of NFRs from concrete requirements specifications, RQ3 is concerned with theory building before we evaluate the resulting theory again in a practical (industrial and academic) context (RQ4).

Please note that in this paper, we provide insights into the current data analysis that concerns RQ1 and RQ2.

### B. Case and Subject Description

The study objects for RQ1 and RQ2 are based on 346 NFRs taken from 11 industrial specifications from 5 different companies for different application domains and of different sizes. The specifications further differ in the level of abstraction, detail, and completeness. As these specifications are confidential, we cannot give detailed information on the individual NFRs nor on the projects. However, Table I provides an overview of the study objects in scope of RQ1 and RQ2, their domain, and exemplary (anonymized) NFRs.

All data classifications are performed independently by two different researchers (1st and 3rd author). Both researchers are working for more than three years in requirements engineering and model-based development research.

<sup>1</sup>Due to space limitations, we do not provide a blue print of the tree.

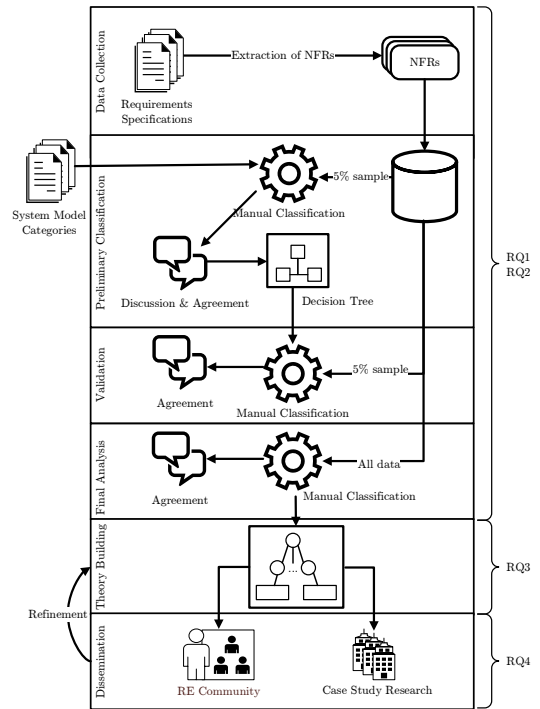


Fig. 1. Overview of the study design and the relation of the RQs to the steps.

### C. Data Collection & Analysis Procedures

We collected all requirements from the specifications that are explicitly labelled as *non-functional*, *quality*, or as one specific class of NFR, e.g. *availability*. To answer RQ1, we classified the resulting NFRs according to the following dimensions:

**Quality characteristic** from the quality model for external and internal quality (ISO/IEC 9126). See [2] for the individual characteristics.

**Scope** of the NFR, i.e., *system embedded in its context, the system itself, or a sub-system*.

For RQ2, we base our classification on one established system modeling theory [7]. Here, we classified the NFRs according to the following fundamental dimensions (see [8]):

**Modeling View**, i.e. does the NFR describe *externally visible system behavior, internal system behavior, or representational aspects*. This dimension differentiates behavior that is externally visible only, also known as black box behavior (see, e.g., NFR of S10, Table I), behavior that is internal to the system, also known as glass box behavior (see ex. NFR of S6, Table I), and representational aspects of a system (see, e.g., NFR of S7, Table I).

**System Modeling Concept**, i.e. does the NFR describe *interface and interface behavior, architecture and architectural behavior, or state and state transition behavior*. This dimension differentiates behavior in terms of interaction over the system boundary (see, e.g., NFR of S10, Table I), structuring the system into a set of sub-systems with their connections and their interactions (see, e.g., NFR of S2, Table I), and describing the state space and state transitions of a system (see, e.g., NFR of S5, Table I).

**Modeling Theory**, i.e., with what means is the NFR described (*syntactical, logical, probabilistic, timed*)? This

TABLE I. OVERVIEW OF THE STUDY OBJECTS FOR RQ1 AND RQ2.

Spec.	Family of Systems <sup>1</sup> (Domain)	# Reqs	# NFRs	% NFRs	Exemplary NFR (anonymized due to confidentiality)
S1	BIS (Finance)	200	61	30.5%	<i>The availability shall not be less than [x]%. That is the current value.</i>
S2	BIS (Automotive)	177	40	22.6%	<i>An online help function must be available. [It] has to be accessible in every dialog. [...]</i>
S3	BIS (Finance)	107	5	4.7%	<i>The maximal number of users that are at the same time active in the system is [x].</i>
S4	ES/BIS (Travel Mngmt.)	38	14	36.8%	<i>The [system] is used by users that are directly in contact with customers. Thus, long response times are not acceptable. The time of [x]% of the functions within the [system]-components shall not be more than [x] seconds.</i>
S5	ES/BIS (Travel Mngmt.)	69	16	23.2%	<i>It must be possible to completely restore a running configuration when the system crashes.</i>
S6	ES (Railway)	35	14	40.0%	<i>The delay between passing a [message] and decoding of the first loop message shall be <math>\leq</math> [x] seconds.</i>
S7	ES (Railway)	122	19	15.6%	<i>The collection, interpretation, accuracy and allocation of data relating to the railway network shall be undertaken to a quality level commensurate with the SIL [x] allocation to the [system] equipment.</i>
S8	ES/BIS (Traffic Mngmt.)	554	128	23.1%	<i>It shall be possible to install programs and configuration data separately.</i>
S9	ES (Railway)	393	12	3.0%	<i>The [system] will have a Mean Time Between Wrong Side Failure (MTBWSF) greater than [x] h respectively a THR less than [x]/h due to the use of [a specific] platform.</i>
S10	ES (Railway)	122	31	25.4%	<i>The [system] system shall handle a maximum of [x] trains per line.</i>
S11	BIS (Facility Mngmt.)	24	6	25.0%	<i>The architecture as well as the programming has to guarantee an easy and efficient maintainability.</i>
$\Sigma$ 11		$\Sigma$ 1.841	$\Sigma$ 346	18.8%	

<sup>1</sup> System classes considered are BIS (Business Information systems) and ES (Embedded Systems) as well as hybrids of both.

dimension distinguishes between NFRs that describe syntactical structure (see, e.g., NFR of S2, Table I), or NFRs that describe behavior. The latter is further refined to the kind of behavior: logical, probabilistic, or timed.

### III. CURRENT STATUS AND PRELIMINARY RESULTS

At the moment of writing this paper, we completed the pre-study based on a sample and reached the validation phase of our study. So far, we analyzed 38 NFRs out of 346 (approx. 10%), created the decision tree, discussed the results, and agreed on the classification. In this section, we will give a brief overview of the preliminary results and analysis.

#### A. Preliminary Results

The results of RQ1 are shown in the following table and the results of RQ2 are shown in Figure 2(a)-(c).

Quality characteristic	count	percentage
Functionality - Security	9	23.7%
Functionality - Suitability	8	21.0%
Portability - Adaptability	5	13.2%
Portability - Installability	3	7.9%
Reliability - Maturity	3	7.9%
Reliability - Recoverability	2	5.3%
Usability - Understandability	2	5.3%
Efficiency - Resource Utilization	2	5.3%
Efficiency - Time Behavior	1	2.6%
Functionality - Accuracy	1	2.6%
Functionality - Interoperability	1	2.6%
Usability - Learnability	1	2.6%
Scope	count	percentage
System in Context	9	23.7%
System	27	71.1%
Sub-system	2	5.3%

#### B. Preliminary Interpretation

Concerning RQ1, one can already see that about 50% of all NFRs in the sample are in a sub-category of *functionality*. Furthermore, within the category *functionality*, most NFRs are concerned with *security* (9) or with *suitability* (8). The latter are classical functional requirements. Moreover within

TABLE II. COHEN'S KAPPA OF 1ST AND 2ND PRELIM. CLASSIFICATION

Category	$\kappa_v$ (1st)	p-val <sub>v1</sub> (1st)	$\kappa_v$ (2nd)	p-val <sub>v2</sub> (2nd)
ISO/IEC 9126	0.577(10)	9.33E-5	0.505(18)	1.65E-11
Scope	0.0(13)	NaN	0.133(18)	0.475
S.M. Concept	0.0(11)	NaN	-0.0263(13)	0.871
View	-0.0263(13)	0.882	0.337(18)	0.0543
Theory	0.0(12)	NaN	-0.111(15)	0.515

the rest, most NFRs are concerned with either portability (8) or with reliability (5). The scope of most requirements is the *system* (71.1%), while only 23.7% describe a functionality of the *system in its context* and only 5.3% describe a functionality of a *sub-system*.

Concerning RQ2, we can see that in particular almost all NFRs within the *Functionality - Security* and *Functionality - Suitability* class describe externally visible behavior, interface and interface behavior, and are described logically. Furthermore, we can see that 68.4% of the NFRs describe externally visible behavior (15.8% internal system behavior and 15.8% representational aspects), 78.1% interface and interface behavior (15.6% architecture and architecture behavior and 6.3% state and state transition behavior), and 78.1% are described logically (28.9% syntactical, 2.6% probabilistic, and 2.6% timed).

In our pre-run using the sample, we did not (yet) analyze further differentiations, e.g., according to the class of systems. The results still already indicate that many NFRs seem to describe externally visible behavior, interface and interface behavior, and they are described logically. This is how classical functional requirements are specified and, furthermore, about 50% of all NFRs in the sample are in the sub-category of *functionality*. This indicates how blurry the borders between functional requirements and NFRs actually are.

### IV. OPEN ISSUES AND THREATS TO VALIDITY

In the course of designing the study and later on during initial classifications based on the sample, we were confronted with different issues of which some still remain open. In this

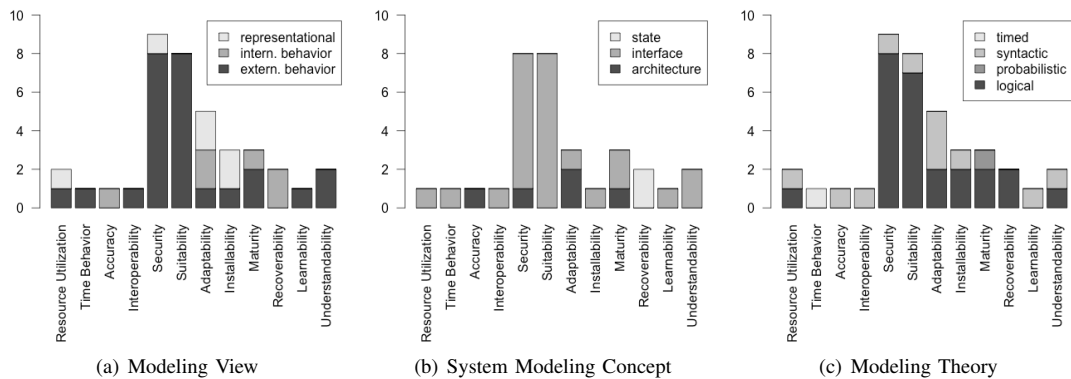


Fig. 2. Results for RQ2: Distribution of the ISO quality attributes among the system modeling dimensions.

section, we provide an overview of those issues we consider to result in the biggest threats to the validity of our study.

**Data Representativeness.** We see the biggest threat to the validity to be in the representativeness of the data on which we built our analysis. The concerns range from the representativeness of the way the NFRs are specified to the completeness of the data as it currently only covers the particularities of selected industrial contexts only.

**NFR Selection.** We only collected the requirements that are explicitly labelled as non-functional or quality. With this selection procedure, some relevant NFRs may be missed or irrelevant ones may be included. To address this threat, we plan to perform the classification on the whole data set as future work (including functional and non-functional requirements).

**Classification Dimensions.** To answer RQ1 and RQ2, we classified our data based on multiple dimensions. One open issue concerns the validity of those dimensions themselves. The fuzziness of the dimensions manifests itself in the low inter-rater agreement and low kappa values (see Table II) which was also the reasons for us to elaborate a decision tree. Yet, another reason for the disagreement was that the NFRs were analyzed in isolation and that they often do not provide sufficient information to understand them without the necessary context. Finally, the third problem affecting the classification is given by the ISO/IEC classification itself which we took as a reference and which doesn't provide sufficient guidance for the classification; for instance if considering *Reliability-Maturity* and *Reliability-Fault Tolerance*. Maturity is defined as *the capability of the software product to avoid failure as a result of faults in the software* and fault tolerance as *the capability of the software product to maintain a specified level of performance in cases of software faults or of infringement of its specified interface*. The definitions of both quality attributes are very close and, thus, the classification of exemplary NFR of S9 (see Table I) resulted in *Maturity* from one classifier and *Fault Tolerance* from the other.

**Contextualization.** The quality of our study is very much dependent on the possibility to reproduce the results, which in turn is dependent on the clearness of the context information. The latter, however, is strongly limited by NDAs that too often prevent providing full disclosure of the contexts and even the project characteristics.

## V. CONCLUSION

The main goal of this short paper was to initiate discussions on the issues related on the integration of NFRs in seamless modeling in general and, in particular, discussions on open methodological issues of our study.

To this end, we presented our overall study design which includes a multi-staged procedure to infer an empirically founded theory on specifying NFRs to support seamless modeling. Then, we discussed preliminary results from a sample (approx. 10%) and current open issues and threats to validity of our study. The preliminary results already indicate to interesting points; for example, many of commonly used NFR classes concern system facets in a way that shows how blurry the borders between functional and non-functional requirements are. Furthermore, we identified fields of improvement for our study, for example, the low inter-rater agreement during the classification.

We conclude so far that our overall study design seems suitable to obtain the envisioned theory in the long run, but we could also show current issues that are worth discussing within the empirical software engineering community.

## REFERENCES

- [1] M. Broy, M. Feilkas, M. Herrmannsdoerfer, S. Merenda, and D. Ratiu, "Seamless model-based development: From isolated tools to integrated model engineering environments," *Proc. of the IEEE*, vol. 98, 2010.
- [2] ISO, "ISO/IEC 9126-1:2001, Software engineering – Product quality – Part 1: Quality model," Tech. Rep., 2001.
- [3] M. Glinz, "On non-functional requirements," in *RE*. IEEE, 2007.
- [4] L. Chung and J. C. S. do Prado Leite, "On non-functional requirements in software engineering," in *Conceptual modeling: Foundations and applications*. Springer, 2009.
- [5] D. Ameller, X. Franch, C. Gómez, J. Araujo, R. Svensson, S. Biffi, V. Cabot, J. Cortellessa, M. Daneva, D. Méndez Fernández, A. Moreira, H. Muccini, A. Vallecillo, M. Wimmer, V. Amaral, H. Brunelière, L. Burgueño, M. Goulão, B. Schätz, and S. Teufl, "Handling Non-Functional Requirements in Model-Driven Development: An Ongoing Industrial Survey," in *RE*, 2015.
- [6] R. B. Svensson, T. Olsson, and B. Regnell, "An investigation of how quality requirements are specified in industrial practice," *IST*, vol. 55, 2013.
- [7] M. Broy and K. Stølen, *Specification and development of interactive systems: focus on streams, interfaces, and refinement*. Springer, 2012.
- [8] M. Broy, "Rethinking Nonfunctional Software Requirements," *IEEE Computer*, vol. 48, no. 5, 2015.