Non-Functional Requirements as Qualities, with a Spice of Ontology

Feng-Lin Li[1], Jennifer Horkoff[1], Alexander Borgida[2], Renata Guizzardi[3], Giancarlo Guizzardi[3], John Mylopoulos[1], Lin Liu[4]

[1] University of Trento, Trento, Italy
[2] Rutgers University, New Brunswick, USA
[3] Federal University of Espírito Santo (UFES), Vitória, Brazil
[4] Tsinghua University, Beijing, China
2014-08-29

Outline

- **Motivation**

- **NFRs as Requirements over Qualities**

- **A Framework for Goal Models with Qualities**

- **Evaluation**

- **Conclusions and Future Work**

Non-functional requirements (NFRs)

- **No agreement on what they are, e.g.**
  - ✓ *How well a system performs its functions*
  - ✓ *Anything that is not functional* **[Chung09]**
  - ✓ *Attributes or constraints of a system* **[Glinz07]**
- **What about**
  - ✓ **"software maintainers shall be able to integrate new functionalities within one work day"**
  - ✓ **"all updates to databases shall be performed by the application"**

NFRs are often problematic

- Because of their vague, informal nature, e.g.,

  - ✓ (*Vague*) NFR-1: "The product shall return (file) search results in an *acceptable* time."

  - ✓ (*Make-or-break*) NFR-2: "Administrator shall be able to activate a pre-paid card via the Administration section *within 5 sec.*"

  - ✓ (*Practically unsatisfiable*) NFR-3: "The website shall be available for use *24 hours per day, 365 days per year*."

  - ✓ (*Subjective*) NFR-4: "The interface shall be *appealing* to callers and supervisors."

Quality according to ontologies

- **Unified Foundational Ontology (UFO) [Guizzardi05]**

- **"A basic perceivable or measurable characteristic that inheres in and existentially depends on its subject"**

- **Qualities as *mappings***

  - **A quality maps its subjects to values in a quality space**

Quality mapping

trip#1   Cost#1   1000€   Region (low)

Subject domain   Quality space (co-domain)
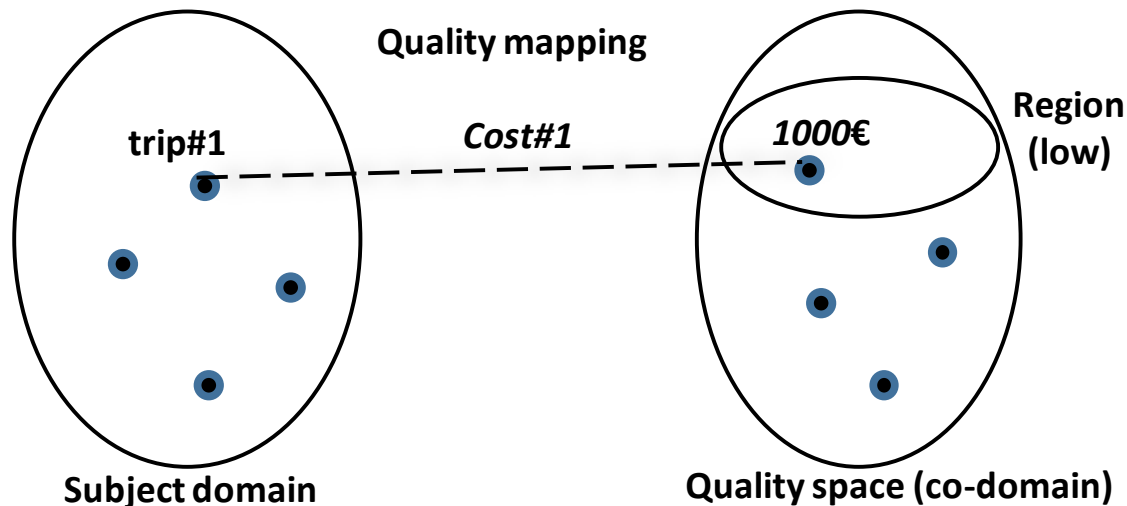
NFRs as requirements over qualities

- An NFR constrains a quality mapping *Q* to take values in a region *QRG* of its quality space for its subject type *SubjT*

- Model NFRs as *Quality Goals (QG)/Constraints (QC)*

  - **QG := Q(*SubjT*) : QRG**

  - **∀*x*. instanceOf (*x*, SubjT) → subregionOf (Q(*x*), QRG)**

> *Examples (NFR-1): The processing time of file research shall be acceptable.*
> *QG1:= processing time (file search): acceptable.*
> *QC1:= processing time (file search): ≤ 8 sec.*
> *QC1 is-operationalization-of QG1*
> *QGi := understandability ({the interface}): intuitive*

Composite NFRs

- **NFRs with qualified subjects**
  - ✓ *SubjT := SubjT <attribute: filler>**
  - ✓ *filler := atomic value | SubjT*

Example 2 (NFR-2): Administrator shall be able to activate a pre-paid card via the Administration section within 5 sec.

activate p-card' :=

> *activate pre-paid card <actor: Administrator>*
> *<means: via the Administration section >.*

QG2 := processing time (activate p-card'): within 5 sec.

Meta-qualities

- **Many requirements have the form ∀x P(x)**

  e.g., "For every request a meeting shall be scheduled"

  "Every file search will be within 5 sec"

- **Quality of fulfillment**

  ✓ *Universality (U)*: degree to which the set of all *x* satisfies *P*

  ✓ *Gradability (G)*: degree to which *P* holds for each *x*

  ✓ *Agreement (A)*: degree to which observers agree that *P* holds for each *x*

Universality as a meta-quality

- **Universality**
  - **U : *power-set*(*SubjT*) → *Percentage***
  - **Input: set of requirement subject instances**
  - **Output: percentage of the instances for which requirement is fulfilled**

Example 3 (NFR-3): The website shall be available for use at 99.5% of the time units in a year.

theWebsite' := theWebsite
   <at: time units <in-period: a year>>
QG3 := availability(theWebsite'): 100%   //the entire unit
QG3-1 := U (QG3): 99.5%   //99.5% of the units in a year

Gradability as a meta-quality

- **Gradability**
  - **G : *SubjT* | *power-set*(*SubjT*) → *Degree of Fulfillment***
  - **Input: a singleton requirement (can also be a set of requirement instances)**
  - **Output: degree of fulfillment on a linear scale [0% , 100%]**

> *QG2:= processing time (activate p-card'): within 5 sec.*
> *QG2-1 := G (QG2): nearly*
> *QG2-2 := G (QG2): 90%*
>
> *QG1:= processing time (file search): acceptable.*
> *QG1-1:= G (QG1): moderately.*

Agreement as a meta-quality

- **Agreement**
  - ✓ **A : (*Requirement*) *SubjT → Ratio***
  - ✓ **Input: a singleton requirement**
  - ✓ **Output: a *ratio* of observers from a given pool who agree that the requirement is satisfied**

$QG4:= look (\{the\ interface\}): appealing$
$QG4\text{-}1:= A (QG4): 80\%\ of\ the\ callers\ and\ supervisors$

Composing meta-qualities

- **Composition**
  - ✓ **U(G(…)) e.g., "*95%* of the activations happen *approximately* within 5 sec.***"**
  - ✓ **A(G(…)) e.g., "*80% of the users* agree the website is rather easy to understand"**
  - ✓ **G(U/A(…)) e.g., "*nearly 90%* of activations take 5 *sec.*, *nearly 80% of the users* report the interface is simple**
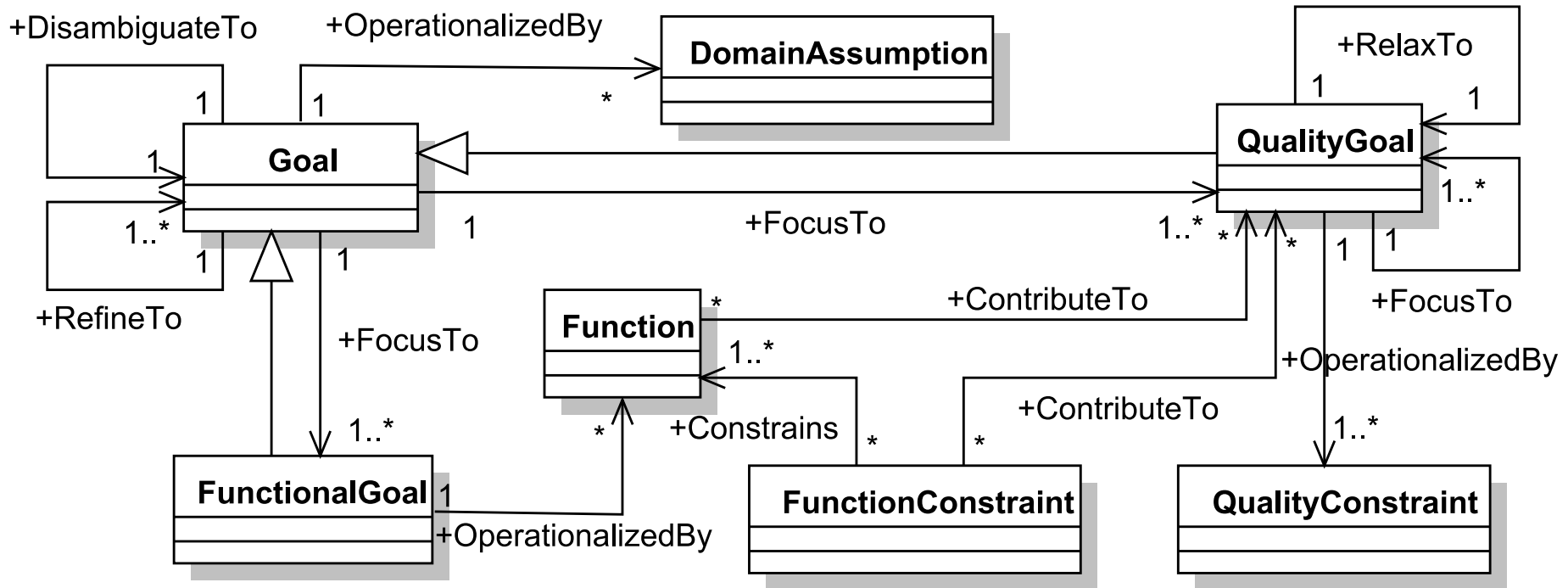 - ✓ **G(U(G(…))) e.g., *nearly 90%* of activations take *nearly* 5 sec.).**

QG2 := processing time (activate p-card'): within 5 sec.
QG2-1 := G (QG2): nearly
QG2-3 := U (QG2-1): 95%

A framework for goal models with qualities

Meta-model

Goal modelling process with qualities
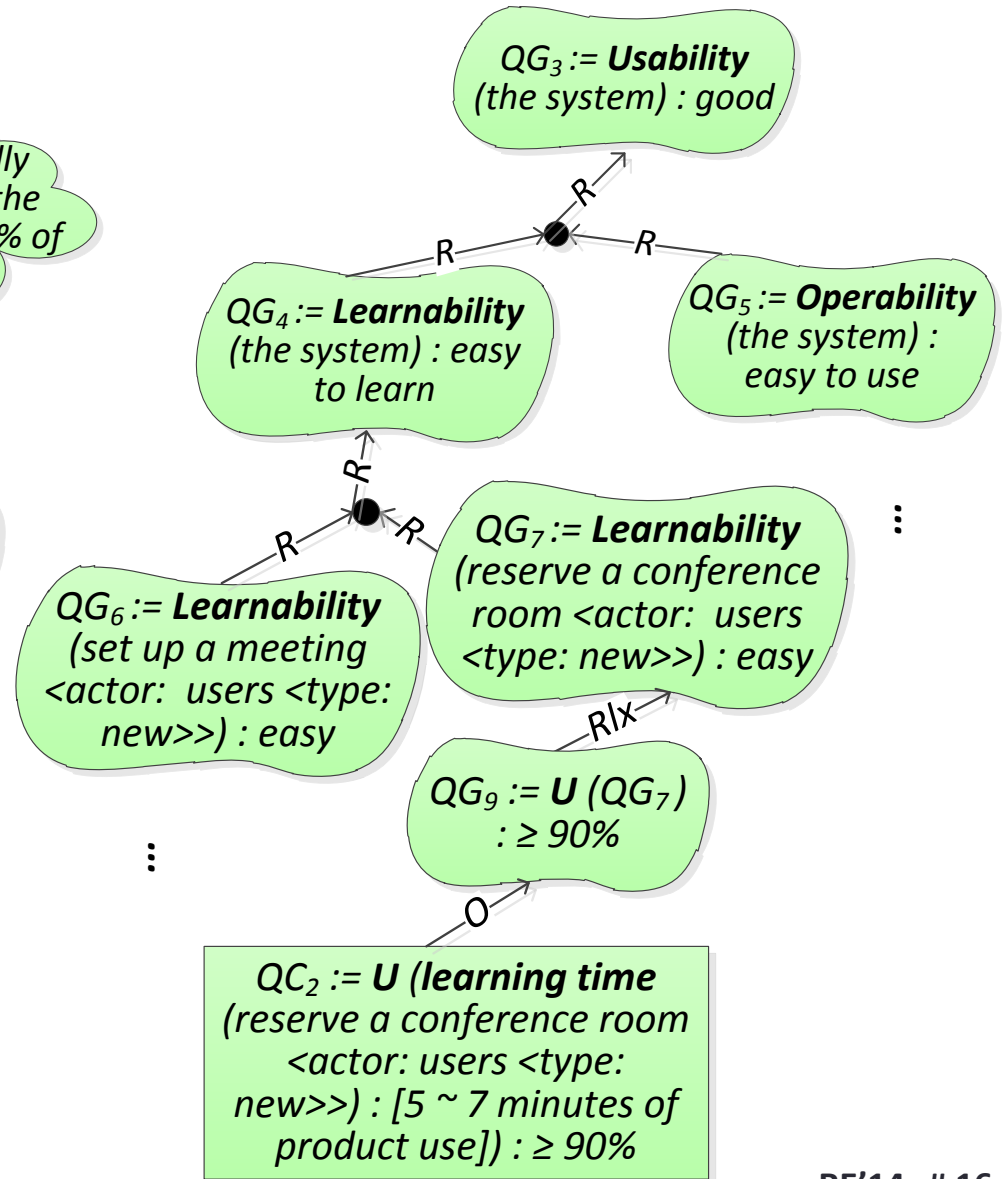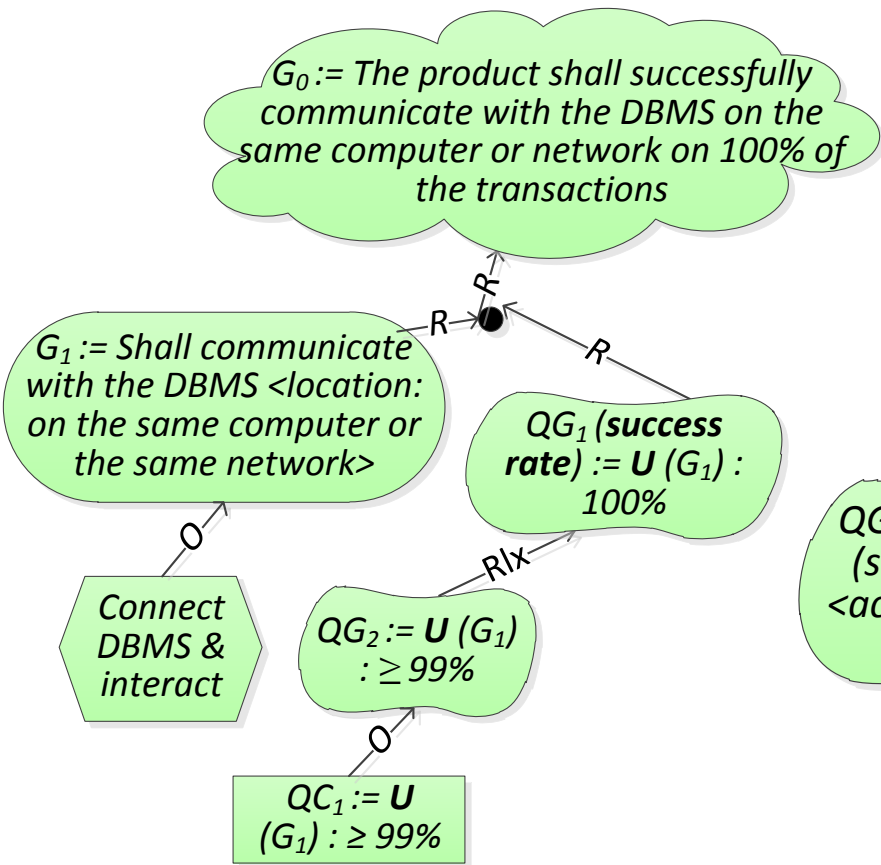
- **Iteratively ask the questions**
  - ✓**Is a requirement/goal unambiguous?**
  - ✓**Is it (practically) satisfiable?**
  - ✓**How do we make it measurable?**
- **Methodology**
  - ✓*Disambiguation*
    - ✓**Requirement is ambiguous if it has multiple interpretations**
    - ✓**E.g., "interface shall have standard menu buttons for navigation"**
  - ✓*Relaxation*
    - ✓**U, G, A**

Other refinement types

- **Focus – narrow down the subject of a goal**
  - ✓**A goal G can be focused into FG or QG**
  - ✓**A QG can be focused into QGs**
    - ✓**According to the quality hierarchy**
    - ✓**According to the subject hierarchy**
- **Operationalization**
  - ✓**Comparison class: the same subject type**
- **Contribution**
  - ✓**Functional elements contribute to quality goals**

An example

$QG_3 :=$ **Usability** (the system) : good

$G_0 :=$ *The product shall successfully communicate with the DBMS on the same computer or network on 100% of the transactions*

$QG_4 :=$ **Learnability** (the system) : easy to learn

$QG_5 :=$ **Operability** (the system) : easy to use

$G_1 :=$ *Shall communicate with the DBMS <location: on the same computer or the same network>*

QG_1 (**success rate**) := **U** (G_1) : 100%

$QG_6 :=$ **Learnability** (set up a meeting <actor: users <type: new>>) : easy

$QG_7 :=$ **Learnability** (reserve a conference room <actor: users <type: new>>) : easy

Connect DBMS & interact

$QG_2 :=$ **U** (G_1) : ≥ 99%

$QG_9 :=$ **U** (QG_7) : ≥ 90%

$QC_1 :=$ **U** (G_1) : ≥ 99%

$QC_2 :=$ **U** (**learning time** (reserve a conference room <actor: users <type: new>>) : [5 ~ 7 minutes of product use]) : ≥ 90%

—**Rlx**→ Relax
—**R**→ Refine
⟶ AND-Refine
—**O**→ Operationalization

Goal

Functional Goal

Task

Quality Goal

Quality Constraint

Evaluation

- **Case study using the PROMISE requirement set [Menzies12]**
  - ✓**15 projects, 625 Requirements, 370 NFRs (11 categories)**
- **Purpose**
  - ✓**Evaluate the need for our framework by classifying NFRs in the data set**
  - ✓**Evaluate the expressiveness of our framework by applying it to the set of NFRs for a meeting scheduler (from PROMISE)**

Results

- **Classification of the 370 NFRs: QR:187 (51%), FR/CF+QR:61 (16%), FR and CF that contribute to QR: 21+36 (15%)**

NFR Category	Count	QR	FR/CF + QR	FR	CF	FR +CF
Usability	67	47	13+1	5(3)	1(1)	0
Security	66	2	11+3	14(11)	32(32)	4
Operational	62	11	10+2	14	12(3)	6
Performance	54	44	4+1	3(2)	1	1
Look and Feel	38	20	7+2	9(1)	0	0
Availability	21	21	0	0	0	0
Scalability	21	19	0	1	0	1
Maintainability	17	8	5	0	4	0
Legal	13	11	0	2(2)	0	0
Fault tolerance	10	4	2	4(2)	0	0
Portability	1	0	0	0	0	0
Total	370	**187**	**61**	52(21)	50(36)	12

QR: quality requirement; FR: functional requirement; CF: constraint over function

More results

- **Potential application of relaxation operators**
  - ✓**370 NFRs → 481 requirements items**
  - ✓**Practically unsatisfiable:15% (86/567), vague: 25% (143/ 567), measurable: 59% (333/567)**
  - ✓**Implicit operator application**
    - ✓**U: 50, G: 10, A: 16**
  - ✓**Number of requirements that likely need relaxation**
    - ✓**U: 86, G: 476, A: 20**

Satisfaction Type	Count#		Implicit Operator Application	Count#
Ambiguous	5		Universality (U)	50
Unsatisfiable	86		Gradability (G)	10
Vague	143		Agreement (A)	16
Measurable	333			

A small case study

- **Meeting Scheduler**
  - ✓**47 NFRs: 21 QRs, 9 FRs, 14 FR+QR, 2 CF+QR, and 1 DA**
  - ✓**58 QGs from QRs, FR+QR and CF+QR (37 items)**
  - ✓**Rewrite the 58 QGs using our syntax**
  - ✓**Build goal models using our methodology**
    - ✓**The full model: http://goo.gl/AxNjPf**

Evaluation conclusions

- **Different elements of our framework indeed useful**
  - ✓**Quality plays a key role among NFRs in RE practice**
  - ✓**Some NFRs are actually CFs**
  - ✓**Many NFRs are ambiguous, (practically) unsatisfiable, vague, and subjective**
- **Our framework is adequate for covering NFRs in practice**
  - ✓**We have tested the expressiveness of our framework using the meeting scheduling case study**
  - ✓**It is able to support the refinement of requirements into ones that are unambiguous, satisfiable and measurable.**

Conclusions and future work

- **Conclusions**
  - ✓**We adopt an ontological account of NFRs as qualities**
  - ✓**Propose three meta-qualities that account for quality of fulfillment of other requirements**
  - ✓**We propose a language for express NFRs**
  - ✓**We present a methodology for deriving unambiguous, satisfiable, and measurable NFR specifications**
- **Future Work**
  - ✓ **Full syntax and semantics of meta-quality operators**
  - ✓ **Contribution links between functional and quality goals**
  - ✓ **Reasoning with quality goals**

References (partial)

1. [Glinz07] M. Glinz, "On non-functional requirements," RE, 2007.

2. [Chung09] L. Chung, J. do Prado Leite, "On Non-Functional Requirements in Software Engineering," in Conceptual Modeling: Foundations and Applications, Springer, 2009, 363–379.

3. [Guizzardi05] G. Guizzardi, Ontological foundations for structural conceptual models. CTIT, Centre for Telematics and Information Technology, 2005.

4. [Menzies12] T. Menzies, B. Caglayan, H. Zhimin, K. Ekrem, K. Joe, P. Fayola, and T. Burak, "The PROMISE Repository of empirical software engineering data," Jun-2012. [Online]. Available: http://promisedata.googlecode.com.

Acknowledgement

Thank You!

Welcome Questions, Comments and Suggestions !