Analisi di conformità dei requisiti

Simone Vuotto – Università degli Studi di Sassari

PROSSIMO Meeting M12, Sassari - 7 Giugno, 2019

Context and Motivation

Requirements are (informal) descriptions of the **expected behavior** of a system, often expressed in the form of **natural language** sentences.

Context and Motivation

Requirements are (informal) descriptions of the **expected behavior** of a system, often expressed in the form of **natural language** sentences.

Requirement specifications are shared among **many stakeholders** who represent set of individuals with some interest in the system.

Context and Motivation

Requirements are (informal) descriptions of the **expected behavior** of a system, often expressed in the form of **natural language** sentences.

Requirement specifications are shared among **many stakeholders** who represent set of individuals with some interest in the system.

Requirements drive the **design and development life-cycle**. They can determine the **success/failure** of a project.

Context and Motivation

Requirements are (informal) descriptions of the **expected behavior** of a system, often expressed in the form of **natural language** sentences.

Requirement specifications are shared among **many stakeholders** who represent set of individuals with some interest in the system.

Requirements drive the **design and development life-cycle**. They can determine the **success/failure** of a project.

The **assessment** of requirements is an important yet costly and complex task, still largely carried out manually.

Requirements Engineering

The **Requirements Engineering (RE)** research field aims at developing tools and techniques to analyze and handle requirements in a more efficient and automatic way.

Tasks

The RE practice deals with many different tasks:

- Elicitation
- Modeling and Analysis
- Requirements Management
- Validation and Verification
- Etc.

Consistency Checking

Consistency Checking

The task of detecting errors, missing information and deficiencies that can compromise the interpretation and implementation of the intended system behavior.

Consistency Checking

Consistency Checking

The task of detecting errors, missing information and deficiencies that can compromise the interpretation and implementation of the intended system behavior.

The consistency problem can be formulated in many ways. We focus on *inner consistency*, *i.e.* we check for logical errors that prevent the specification to be satisfied by any system.

Consistency Checking

Consistency Checking

The task of detecting errors, missing information and deficiencies that can compromise the interpretation and implementation of the intended system behavior.

The consistency problem can be formulated in many ways. We focus on *inner consistency*, *i.e.* we check for logical errors that prevent the specification to be satisfied by any system.

Example:
R1: When the button is pressed, the red light should be turned on.
R2: When the button is pressed, the red light should be turned off.

Inconsistency Explanation

What about the feedback to the user?
In a big specification document it may be difficult to find the error...

Inconsistency Explanation

What about the feedback to the user?
In a big specification document it may be difficult to find the error...

Inconsistency Explanation

The Task of finding a minimal set of inconsistent requirements that can help the user finding the inconsistency.

Inconsistency Explanation

What about the feedback to the user?
In a big specification document it may be difficult to find the error...

Inconsistency Explanation

The Task of finding a minimal set of inconsistent requirements that can help the user finding the inconsistency.

Given a set of inconsistent requirements R, finds a minimal subset $R' \subseteq R$ that is still inconsistent.

Logic and Formalization

- We started our investigation with Linear Temporal Logic (LTL)
  - It allows to express properties in a rigorous mathematical notation
  - Good expressiveness and reasoning power
  - Many off-the-shelf powerful tools

Logic and Formalization

- We started our investigation with Linear Temporal Logic (LTL)
  - It allows to express properties in a rigorous mathematical notation
  - Good expressiveness and reasoning power
  - Many off-the-shelf powerful tools
- We introduced LTL($\mathcal{D}_C$), an extension of LTL with atomic constraints of the form $x < c$ and $x = c$ (with constant $c \in \mathbb{R}$)

Logic and Formalization

- We started our investigation with Linear Temporal Logic (LTL)
  - It allows to express properties in a rigorous mathematical notation
  - Good expressiveness and reasoning power
  - Many off-the-shelf powerful tools
- We introduced LTL($\mathcal{D}_C$), an extension of LTL with atomic constraints of the form $x < c$ and $x = c$ (with constant $c \in \mathbb{R}$)
- We showed that LTL($\mathcal{D}_C$) can be reconduced to LTL with the help of additional boolean variables and constraints

Logic and Formalization

- We started our investigation with Linear Temporal Logic (LTL)
  - It allows to express properties in a rigorous mathematical notation
  - Good expressiveness and reasoning power
  - Many off-the-shelf powerful tools
- We introduced LTL($\mathcal{D}_C$), an extension of LTL with atomic constraints of the form $x < c$ and $x = c$ (with constant $c \in \mathbb{R}$)
- We showed that LTL($\mathcal{D}_C$) can be reconduced to LTL with the help of additional boolean variables and constraints

Narizzano, M., Pulina, L., Tacchella, A., Vuotto, S. (2018, April). Consistency of property specification patterns with boolean and constrained numerical signals. In NASA Formal Methods Symposium (pp. 383-398). Springer, Cham.

Property Specification Patterns

Property Specification Patterns (PSPs)

PSPs are a collection of **parameterizable, formalism-independent, high-level specification abstractions**. They are templates that can be filled to express common properties.

Property Specification Patterns

Property Specification Patterns (PSPs)

PSPs are a collection of **parameterizable, formalism-independent, high-level specification abstractions**. They are templates that can be filled to express common properties.

PSPs appear like natural language sentences, but their semantics is fixed and they can be directly encoded in a formal specification language.

Property Specification Patterns

Property Specification Patterns (PSPs)

PSPs are a collection of **parameterizable, formalism-independent, high-level specification abstractions**. They are templates that can be filled to express common properties.

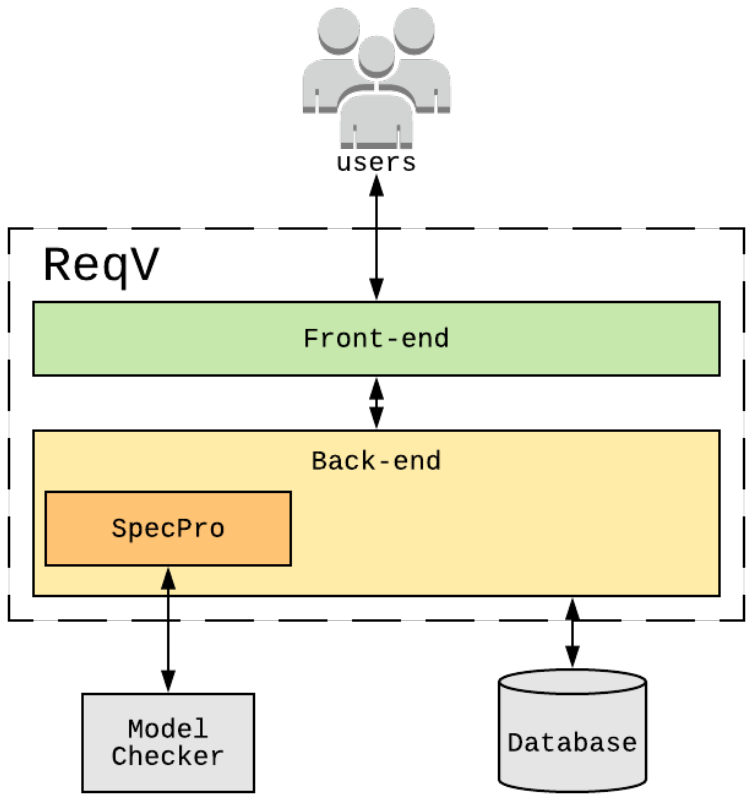
PSPs appear like natural language sentences, but their semantics is fixed and they can be directly encoded in a formal specification language.

Globally, it is always the case that if proximity_sensor < 20 *holds, then arm_idle eventually holds.*

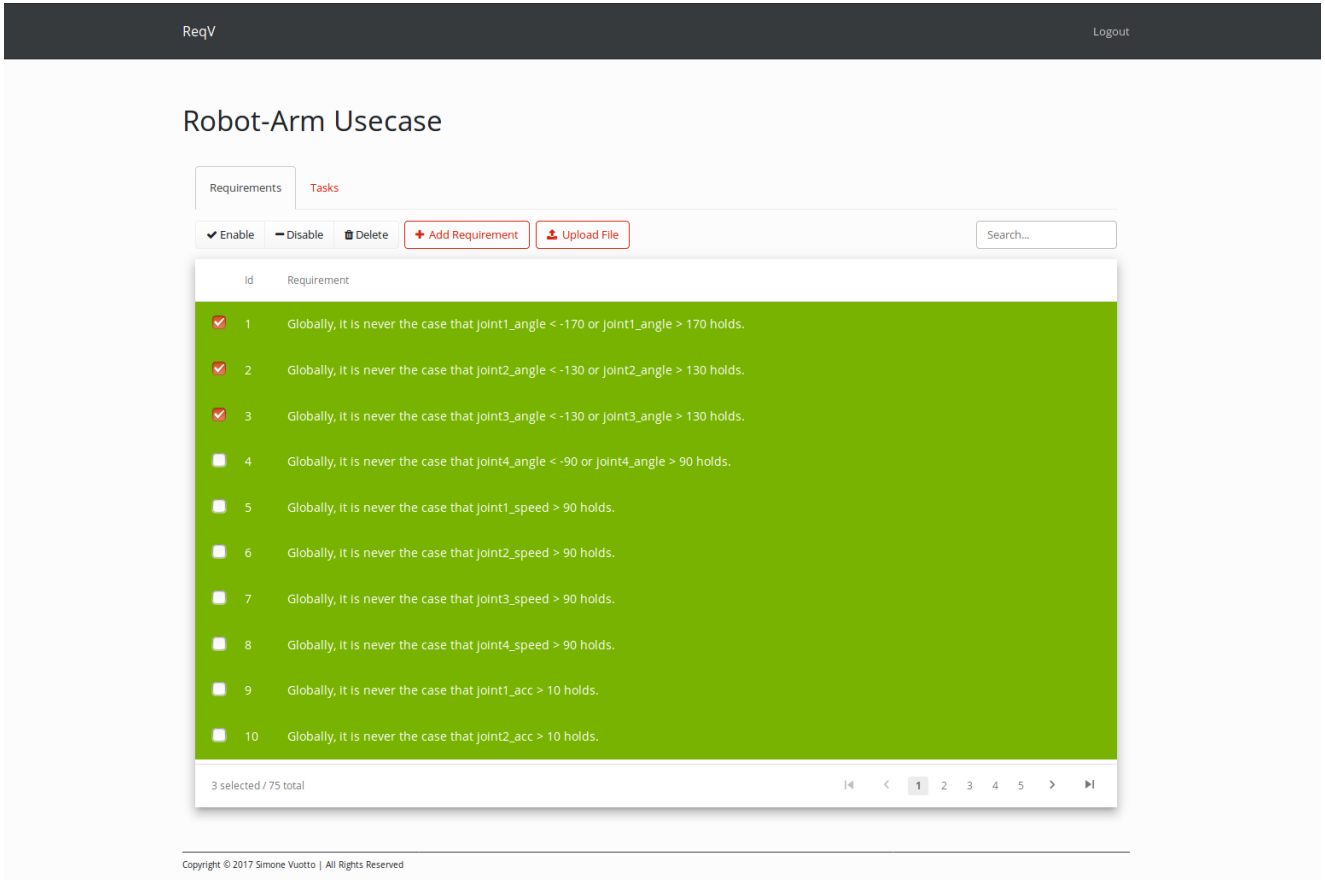
Comparison: LTL vs PSPs

	LTL	PSP
Formal		
Easy to Read		
Easy to Write		
Tools Support		

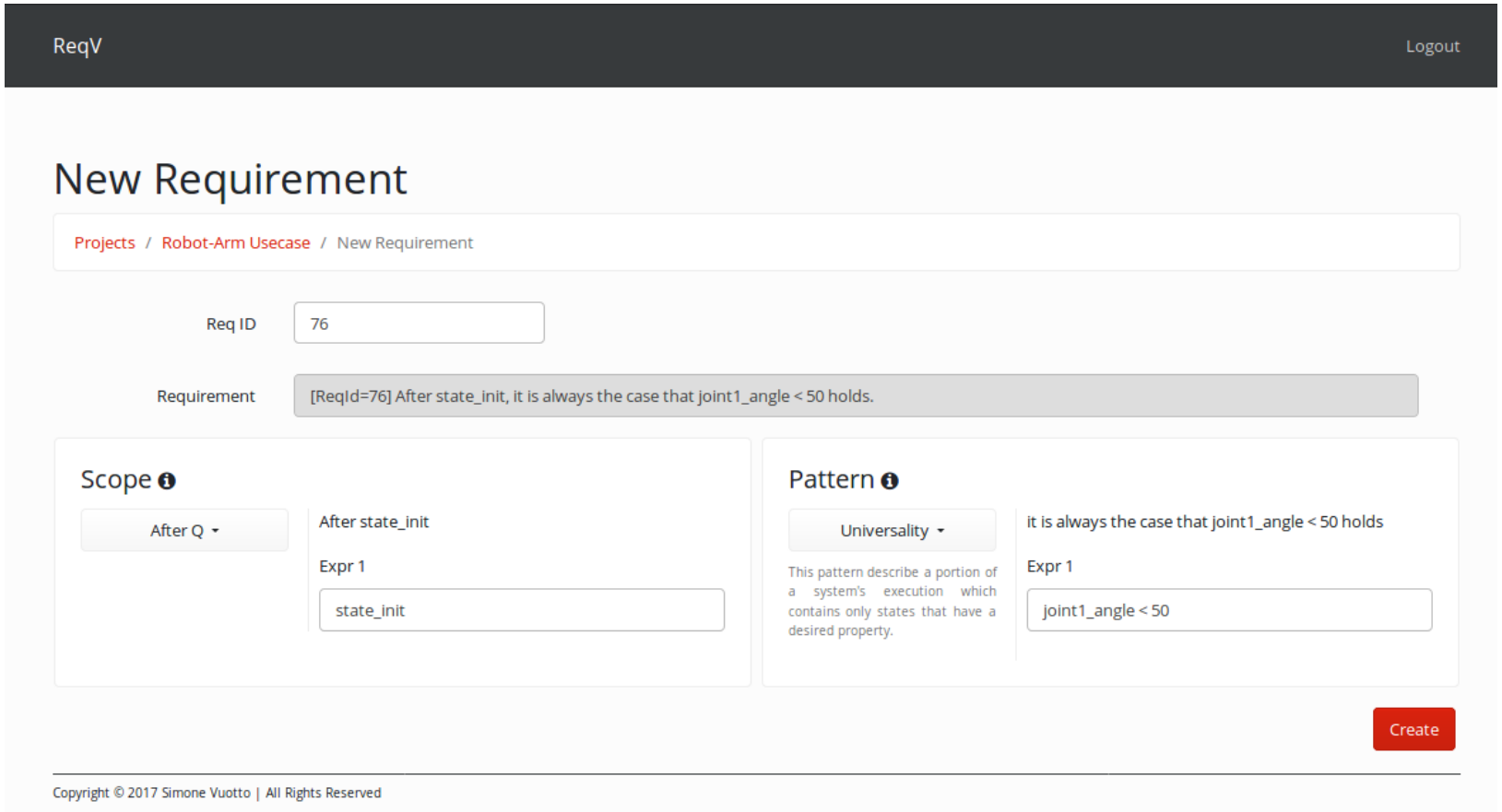
ReqV


users
ReqV
Front-end
Back-end
SpecPro
Model Checker
Database

ReqV 1/2


ReqV
Logout
Robot-Arm Usecase
Requirements
Tasks
Enable
Disable
Delete
Add Requirement
Upload File
Search...
Id
Requirement
1 Globally, it is never the case that joint1_angle < -170 or joint1_angle > 170 holds.
2 Globally, it is never the case that joint2_angle < -130 or joint2_angle > 130 holds.
3 Globally, it is never the case that joint3_angle < -130 or joint3_angle > 130 holds.
4 Globally, it is never the case that joint4_angle < -90 or joint4_angle > 90 holds.
5 Globally, it is never the case that joint1_speed > 90 holds.
6 Globally, it is never the case that joint2_speed > 90 holds.
7 Globally, it is never the case that joint3_speed > 90 holds.
8 Globally, it is never the case that joint4_speed > 90 holds.
9 Globally, it is never the case that joint1_acc > 10 holds.
10 Globally, it is never the case that joint2_acc > 10 holds.
3 selected / 75 total
1 2 3 4 5
Copyright © 2017 Simone Vuotto | All Rights Reserved

ReqV 2/2


ReqV
Logout
New Requirement
Projects / Robot-Arm Usecase / New Requirement
Req ID
76
Requirement
[ReqId=76] After state_init, it is always the case that joint1_angle < 50 holds.
Scope
After Q
After state_init
Expr 1
state_init
Pattern
Universality
This pattern describe a portion of a system's execution which contains only states that have a desired property.
it is always the case that joint1_angle < 50 holds
Expr 1
joint1_angle < 50
Create
Copyright © 2017 Simone Vuotto | All Rights Reserved

Links & References

- Documentation: http://www.sagelab.it/reqv/
- Source Code: https://gitlab.sagelab.it/sage/ReqV
- Video Tutorial: https://youtu.be/2WKSxh64Z2k
- Narizzano, M., Pulina, L., Tacchella, A., Vuotto, S. (2018, April). Consistency of property specification patterns with boolean and constrained numerical signals. In NASA Formal Methods Symposium (pp. 383-398). Springer, Cham.
- Narizzano, M., Pulina, L., Tacchella, A., & Vuotto, S. (2019). Property specification patterns at work: verification and inconsistency explanation. Innovations in Systems and Software Engineering, 1-17.
- Narizzano, M., Pulina, L., Tacchella, A., & Vuotto, S. ReqV: A Tool for Requirements Formal Consistency Checking.
- Vuotto, S. Narizzano, M., Pulina, L., & Tacchella (2019) A. Poster: Automatic Consistency Checking of Requirements with ReqV. In International Conference on Software Testing, Verification and Validation.

Thank you!
Questions?