Requirements-Driven Supervision of Socio-Technical Systems

Davide Dell'Anna

Dept. of Information and Computing Sciences
Utrecht University

REFSQ 2018 - Doctoral Symposium

March 19, 2018

Supervisors
F. Dalpiaz
M. Dastani

RE for Socio-Technical Systems

Traditional Software Systems:

- stable
- largely controllable
- predictable

Socio-Technical Systems:

- **open and dynamic**
- **weakly controllable**
- **unpredictable**

Problem

In their dynamic environment, STSs should comply with requirements.

Self-Adaptive Systems research field: adaptation of systems by alternating different specifications in response to changes in the environment.

But can you change the specification of humans?!

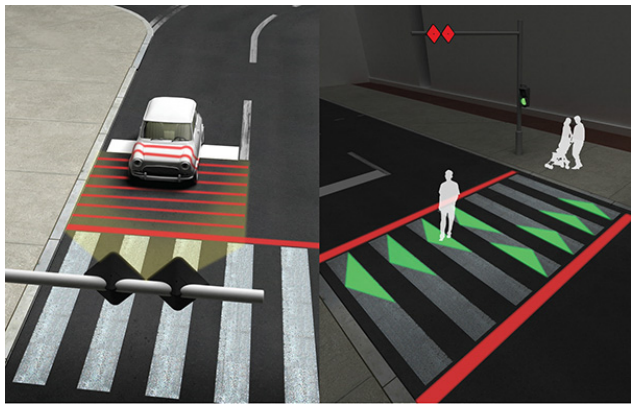
Problem - Smart city example

Municipality **goals**:

- pedestrian safety
- maximize roads' smoothness.

Requirement:
Safe crossings shall be guaranteed

Specification:
Smart zebra lines put in place



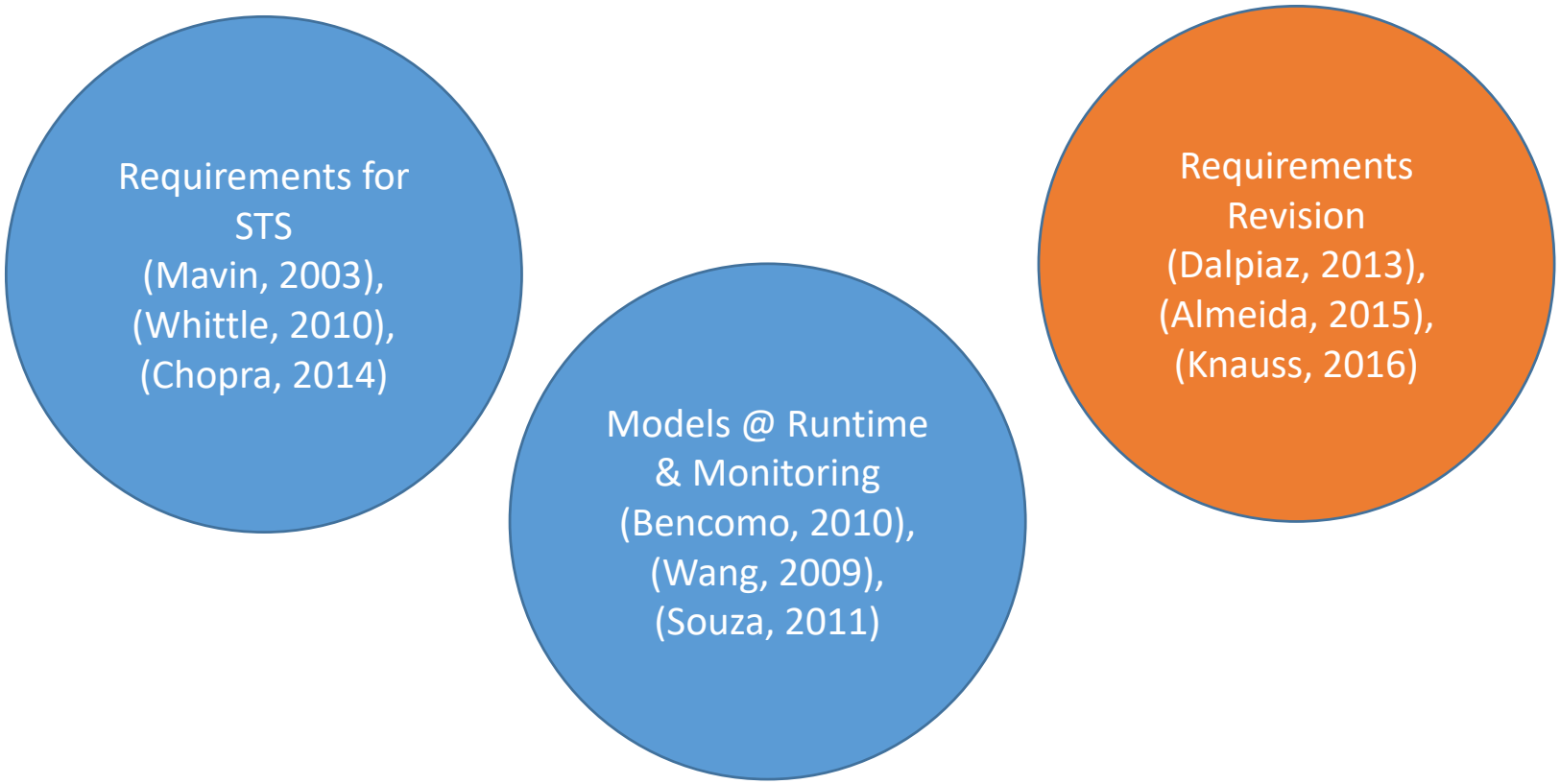
But what if pedestrians cross the street outside the zebra lines?

Problem and Related Work

Problem statement

How to guarantee the achievement of goals of stakeholders of a STS, when the specification of the components cannot be changed?

Baseline and Related Work



Research Question

RQ

How to design and develop a runtime requirements revision framework for Socio-Technical Systems?

Sub-research questions (**SRQs**):

1. What is an expressive, tractable language for specifying requirements for STSs?
2. What are efficient runtime monitoring mechanisms for checking compliance with the requirements represented according to **SRQ 1** ?
3. What are adequate runtime intervention mechanisms to revise the requirements of a STS based on learning from execution data?
4. How to evaluate the effectiveness of the proposal on existing systems?

High-level Solution

Our proposal

A runtime supervision framework:

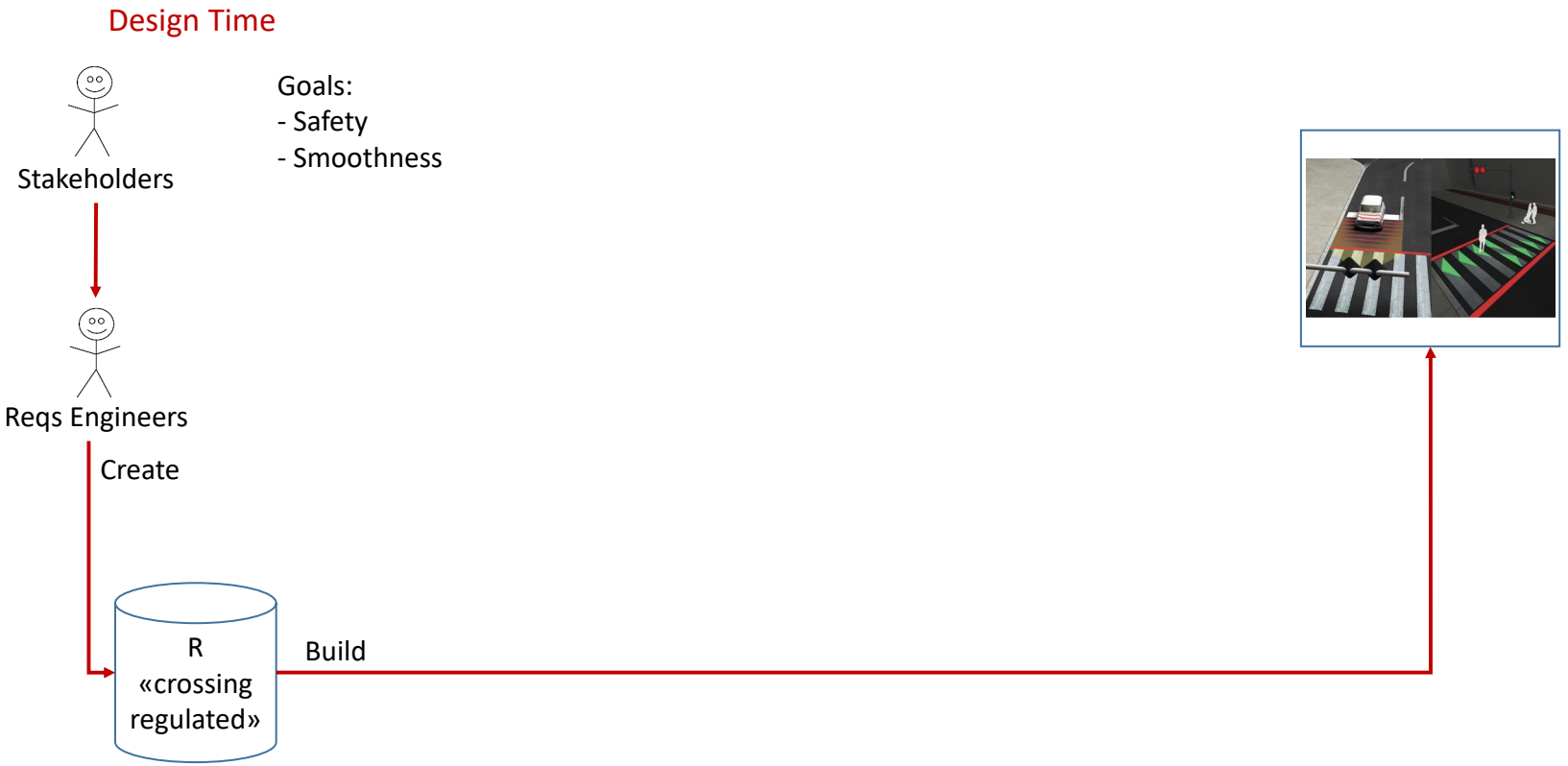
- reifies **goals** and **requirements** at runtime via requirements models
- evaluates the monitored system behavior against the overall goals
- intervenes by deciding how to revise the requirements

Two main pillars:

1. **runtime learning** is necessary to validate the requirement models
2. **requirement revision** is necessary when system adaptation is not possible

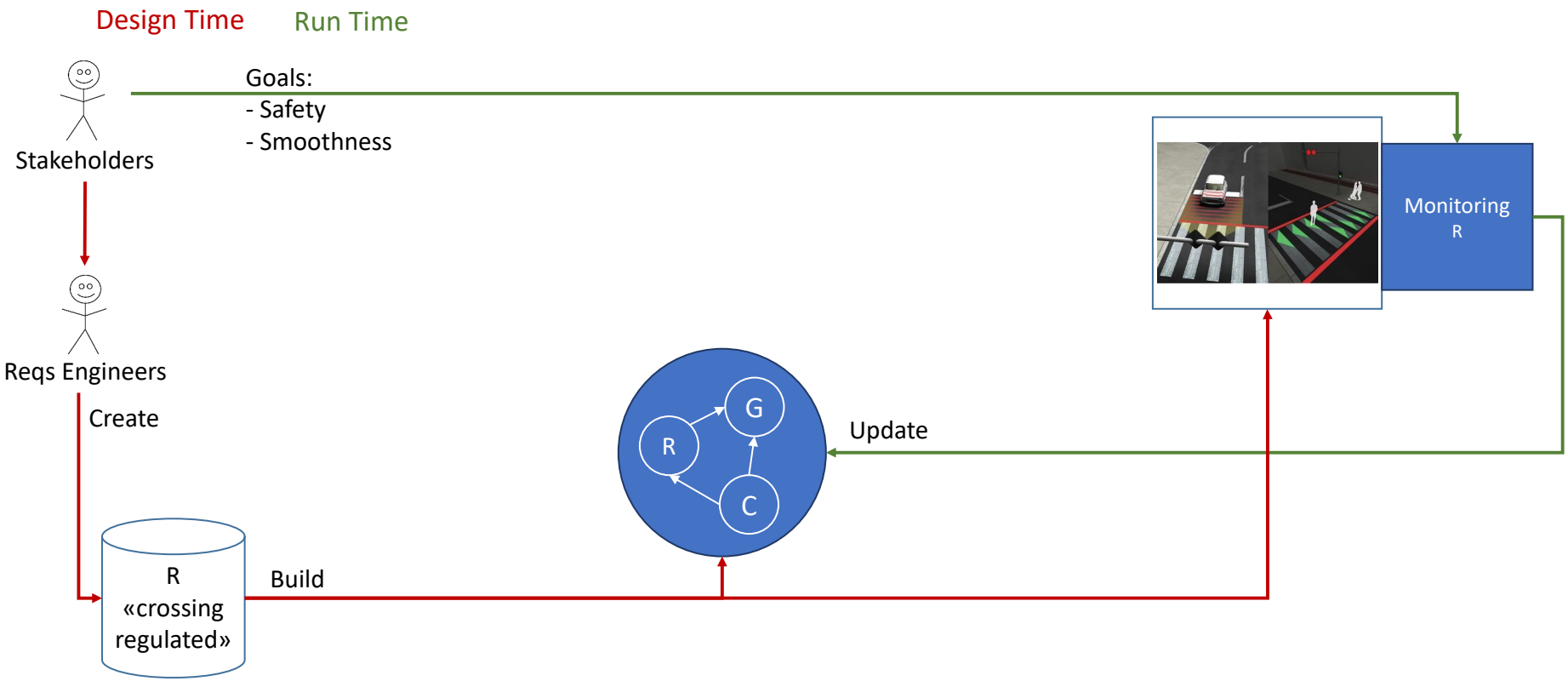
Our runtime supervision framework

Smart city example



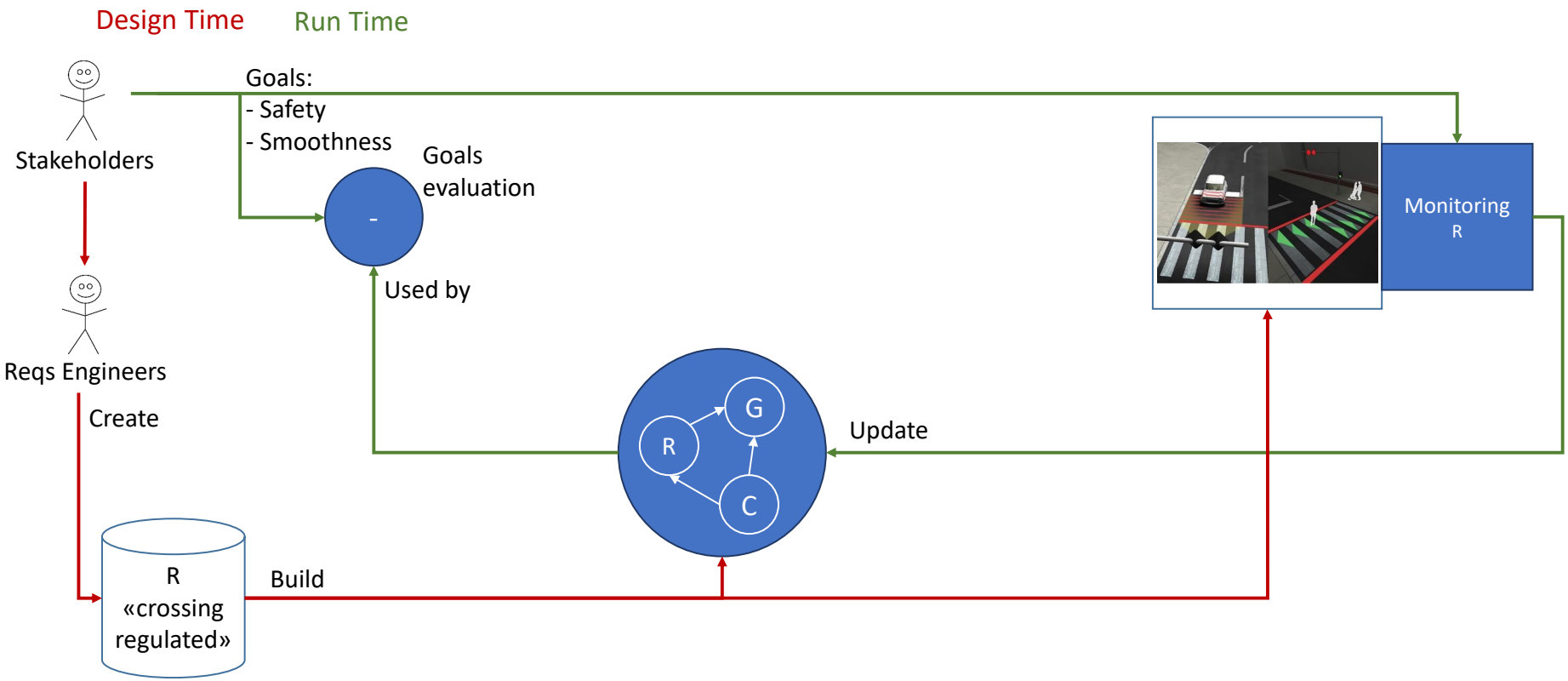
Our runtime supervision framework

Smart city example



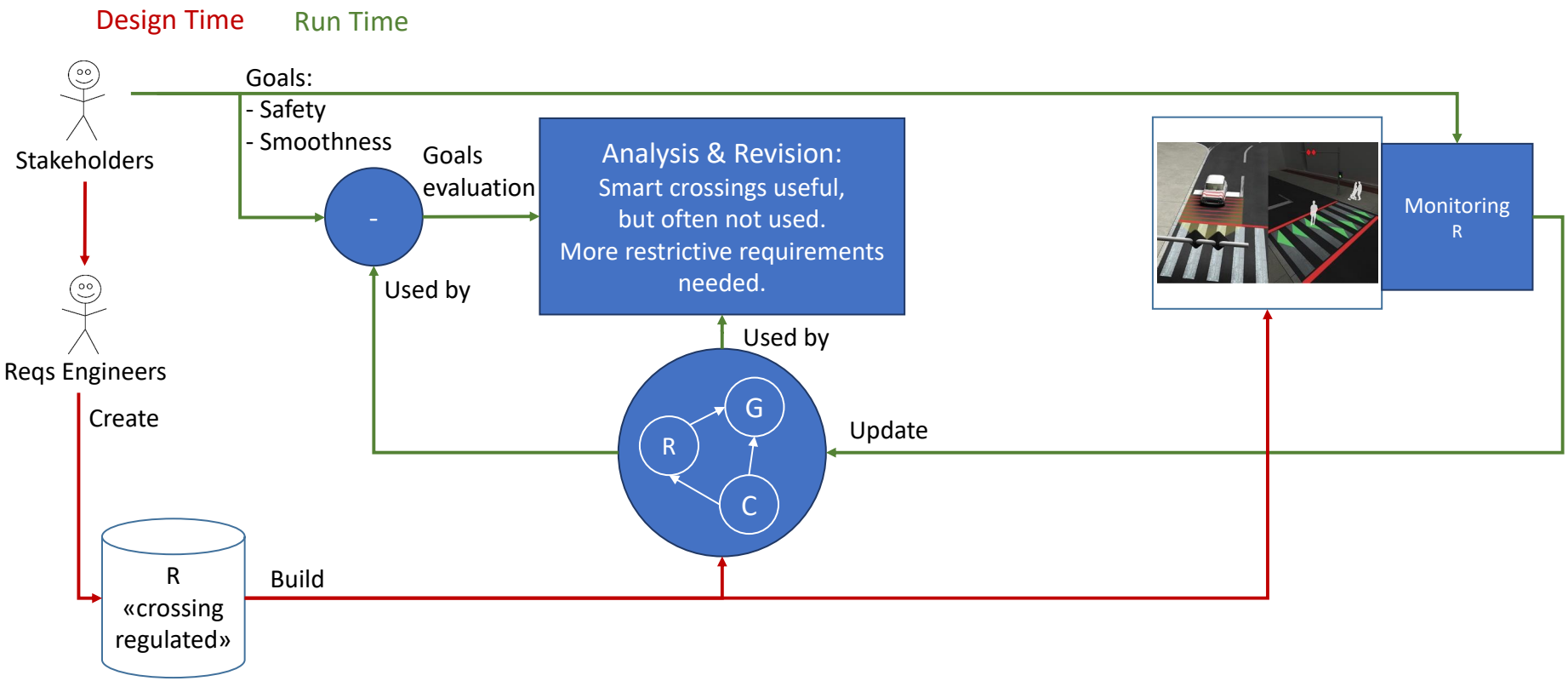
Our runtime supervision framework

Smart city example



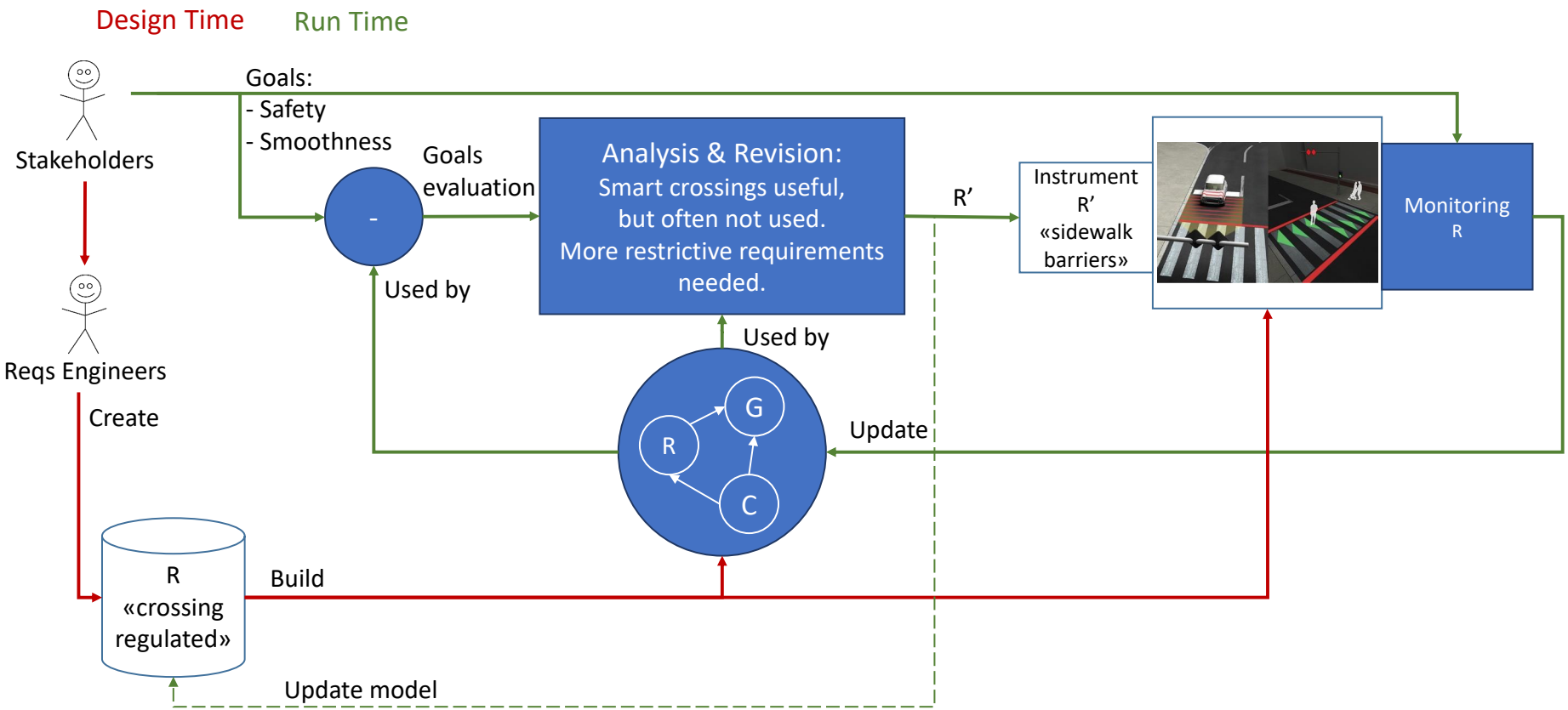
Our runtime supervision framework

Smart city example



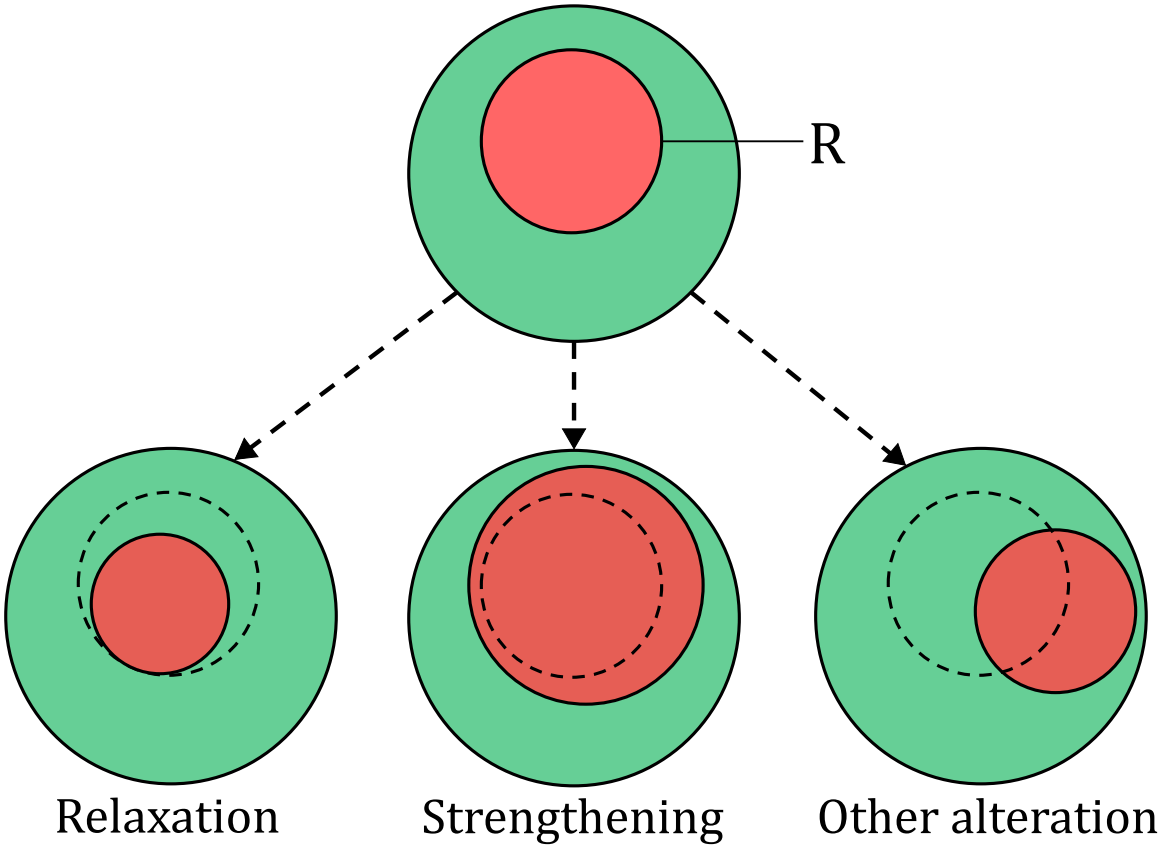
Our runtime supervision framework

Smart city example



Requirements revision

Let $Viol(R)$ be the set of all behaviors of the STS, each violating at least one of the requirements in R.



Behaviors compliance with the requirements:

- Compliant
- Non-compliant

Analyzing System's Behavior

Diagnosis via Bayesian Networks

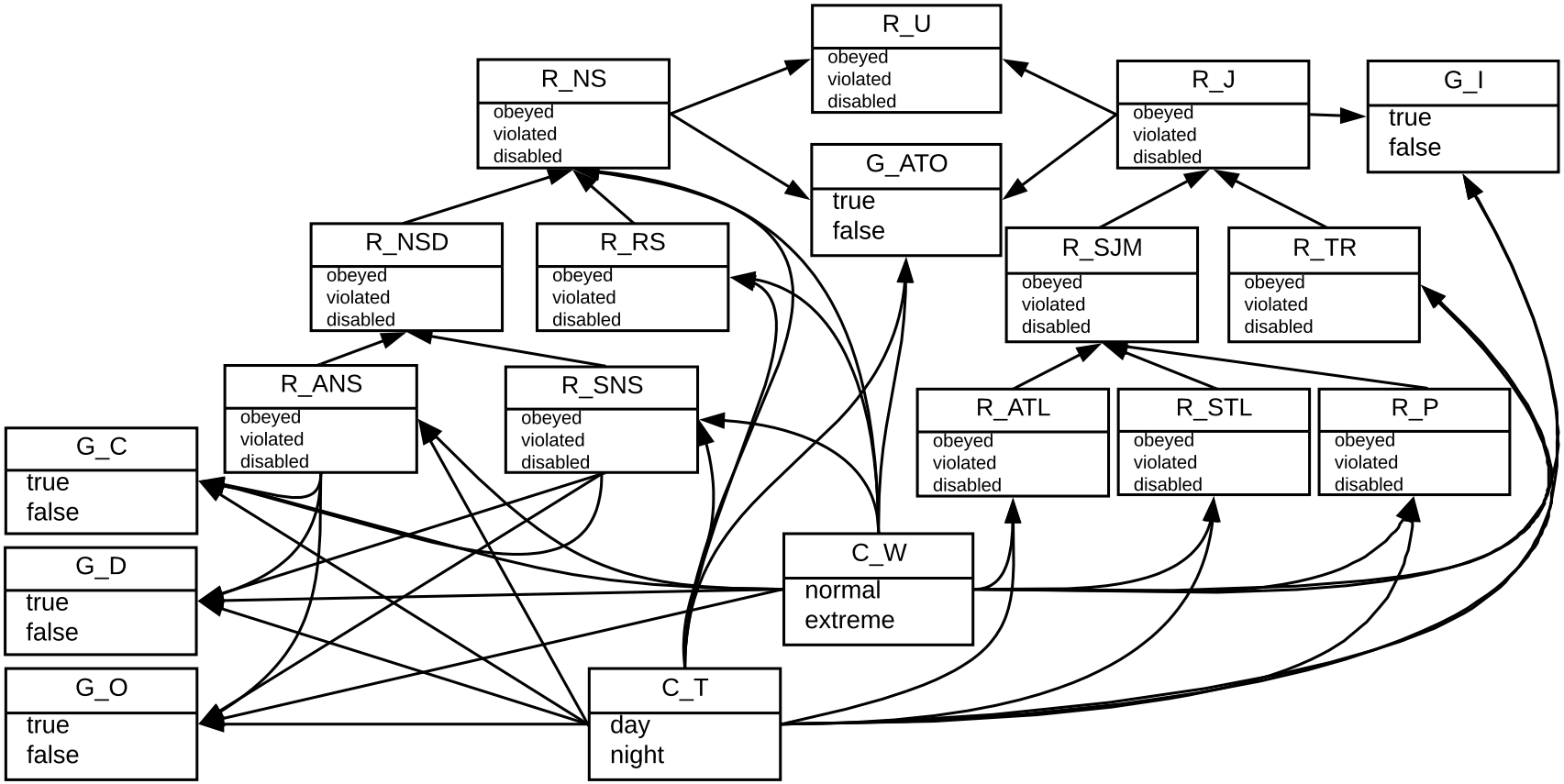


Figure: Bayesian Network collecting data from the STS's execution

Progress and Future Direction

So far.
Focus on **SRQ 3**: **framework**, **learning** and **requirements revision** (Dell'Anna, BNAIC'17), (RE'18 under review).

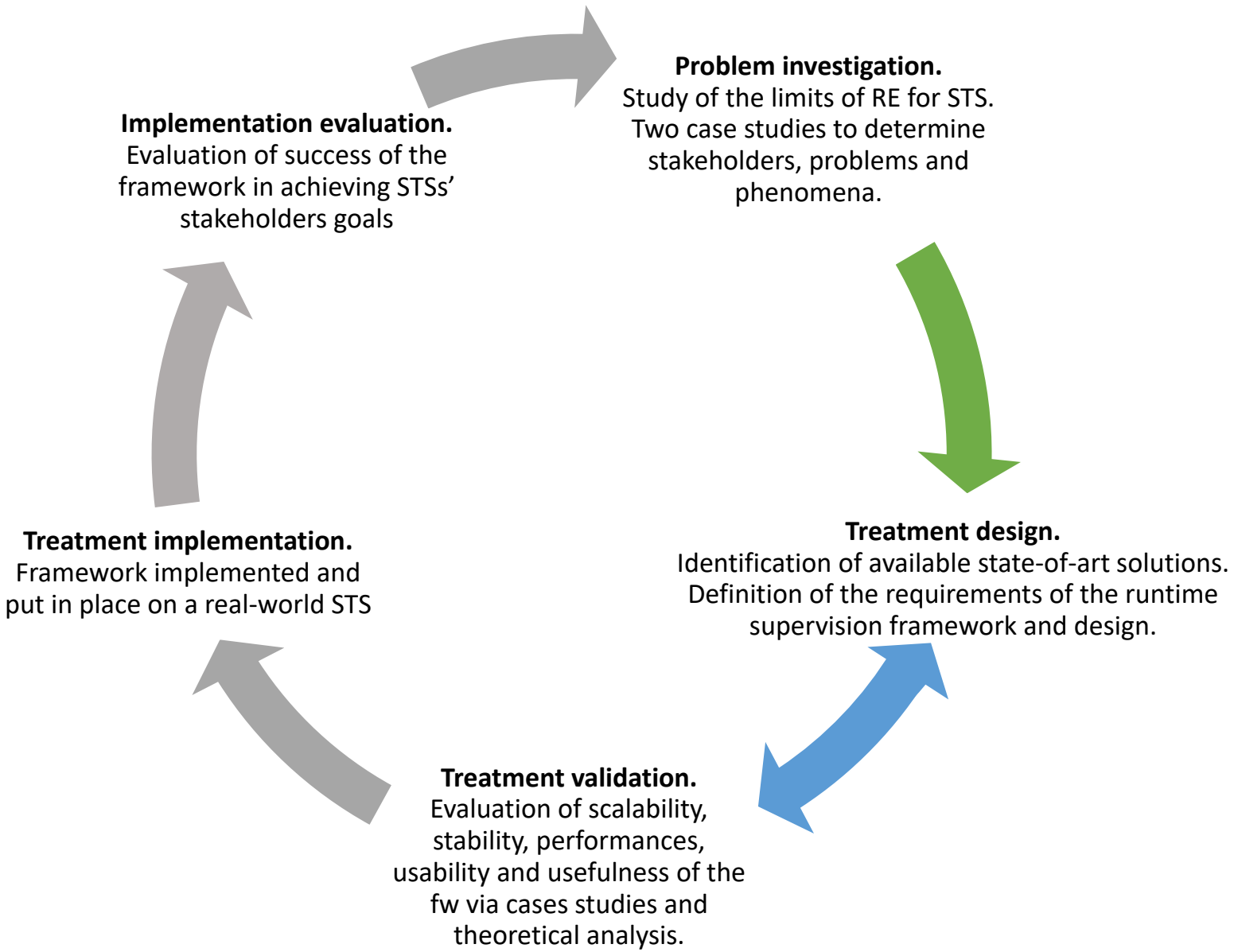
Future direction.
2018-2019

- Identification of a language to express requirements for STSs (**SRQ 1**)
- Monitoring mechanisms for checking reqs compliance (**SRQ 2**)
- Requirements synthesis (extension of **SRQ 3**)

2019-2020

- Formal study of the effects of reqs revision (extension of **SRQ 3**)
- Evaluation on real-world case studies (**SRQ 4**)

Methodology: Design Science



Thank you for your attention.

Davide Dell'Anna
d.dellanna@uu.nl

Research challenges

- **AI vs Requirements Engineering**. Trade-off between theoretical soundness and practical applicability
  - Bayesian Networks can become very large
  - Specify and monitor dynamic requirements
- **Data Collection and Evaluation**.
  - only few related works to compare with: what does *good result* mean?
  - we lack real-world open datasets
  - how to evaluate effectiveness on non-computable goals (e.g., citizen happiness)?