Requirements problem and solution concepts for adaptive systems

Ivan Jureta
Senior Researcher, Fonds de la Recherche Scientifique - FNRS
Associate Professor, Department of Business Administration, University of Namur
ivan.jureta@unamur.be

Goal of the talk is to discuss this:

- How do adaptive systems RE and traditional RE differ?

- Can we use existing requirements modelling languages for adaptive systems RE?

Terminology

RE: Requirements Engineering.

AS: Adaptive System.

ASRE: RE for adaptive systems.

RML: Requirements Modelling Language, a formalism that helps

- **Modelling**: Representation of requirements
- **Reasoning**: Drawing conclusions about requirements
- **Advising**: Recommending next steps when searching for a solution

Overview

1. Key ideas in **ASRE**
2. Standard **RE** problem & solution
3. Minimal **RE** problem & solution
4. Minimal **ASRE** problem & solution
5. Next steps

Key ideas in ASRE (1/2)

1. **(Monitoring) Monitor requirements satisfaction on a running system**
   - Fickas & Feather. Requirements monitoring in dynamic environments.
   - Feather et al. Reconciling system requirements and runtime behavior.
   - Robinson. A requirements monitoring framework for enterprise syst.

2. **(Feedback loops) If requirements are not satisfied (enough), then change AS behavior**
   - Same references as above.

3. **(Probabilistic relaxation) During ASRE, design the specification, so that it maximizes the probability that requirements are satisfied**
   - Letier & van Lamsweerde. Reasoning about partial goal satisfaction...

Key ideas in ASRE (2/2)

4. **(Fuzzy relaxation) Allow requirements to be satisfied to different degrees, at different times, instead of asking for binary satisfaction**
   - Whittle et al. RELAX: a language to address uncertainty...
   - Baresi et al. Fuzzy goals for requirements-driven adaptation.

5. **(Evolution requirements) Requirements stating how to change system behavior depending on the satisfaction of other requirements**
   - Souza et al. Requirements-driven software evolution.

Standard RE problem & solution

Standard **RE problem concept**:

Given a set R of requirements, and a set K of domain knowledge, find a specification S such that K, S |-- R, and that K and S are consistent.

Standard **RE solution concept**:

Specification S that satisfies the conditions in the standard RE problem.

Source: Zave & Jackson. Four dark corners of requirements engineering.

Example

Requirements (R) statements

- Respond to emergency call
- Identify incident location
- Fill out incident report

Domain knowledge (K) statements

- All calls are switched to the dispatch center (no calls are dropped)
- Callers report imprecise incident location

Specification (S) statements

- Implement searchable map in dispatch software
- Fill out incident report form via dispatch software
- Report warning if log receives two or more incidents with same location, within a 5 minute window

Minimal RE problem & solution (1/2)

Standard RE problem is **minimal**, in the sense that if something is removed from it, the remainder is not a problem to solve.

What could we remove?

1. Requirement, domain knowledge, and/or specification concepts
2. Condition that S should be such that S, K |-- R
3. Condition that K and S should be consistent

Minimal RE problem & solution (2/2)

1. After removing the domain knowledge concept, for example:

   Given a set R of requirements, ~~and a set K of domain knowledge,~~ find a specification S such that K, S |-- R, and that K and S are consistent.

2. After removing the condition that S should be such that S, K |-- R:

   Given a set R of requirements, and a set K of domain knowledge, find a specification S such that ~~K, S |-- R, and that~~ K and S are consistent.

3. After removing the condition that K and S should be consistent:

   Given a set R of requirements, and a set K of domain knowledge, find a specification S such that K, S |-- R~~, and that K and S are consistent~~.

In ASRE, are we still trying to solve the minimal RE problem?

No, because the solution is **not** the specification S, but something else.

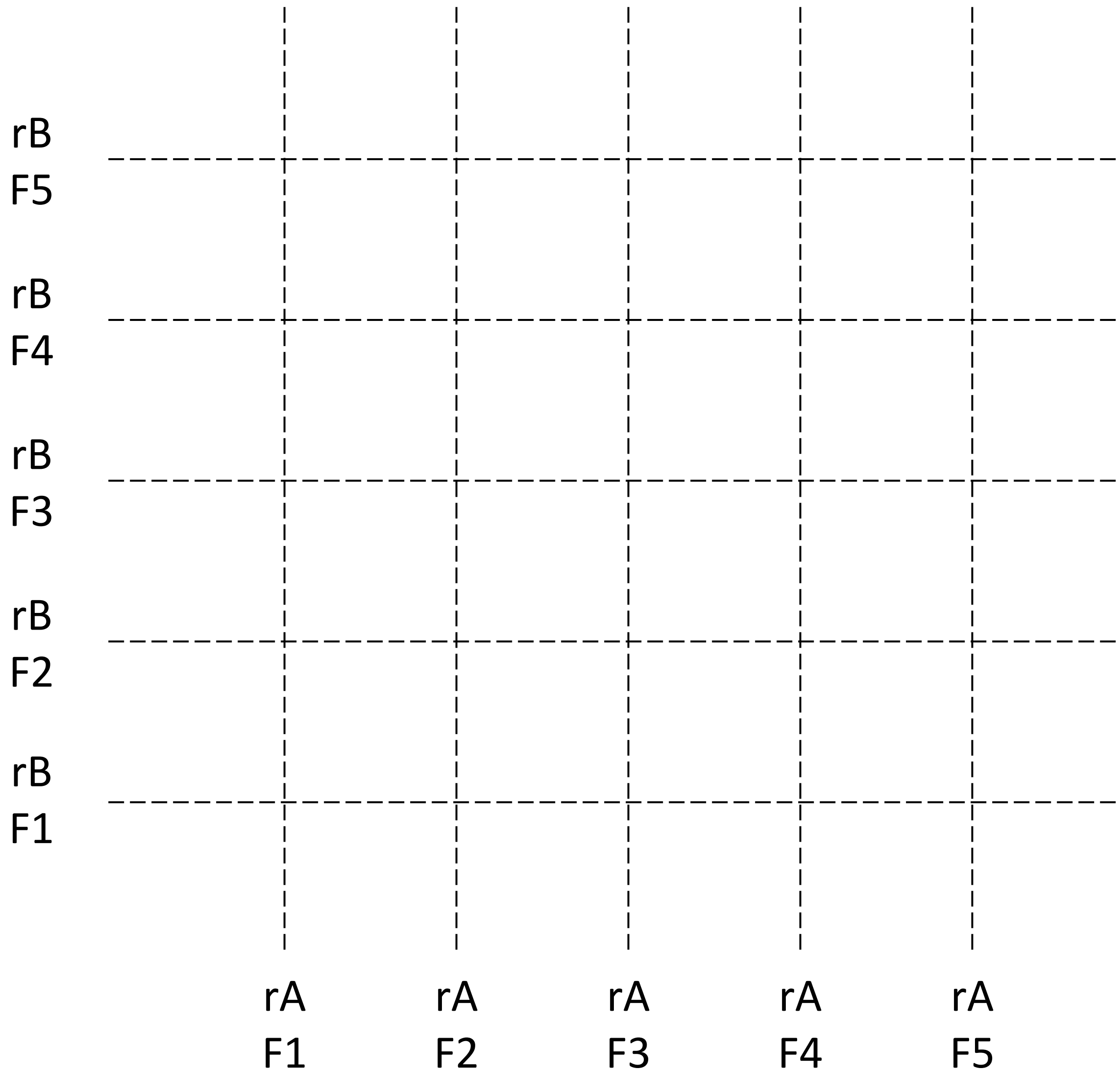
What else?

Consider a trivial example:

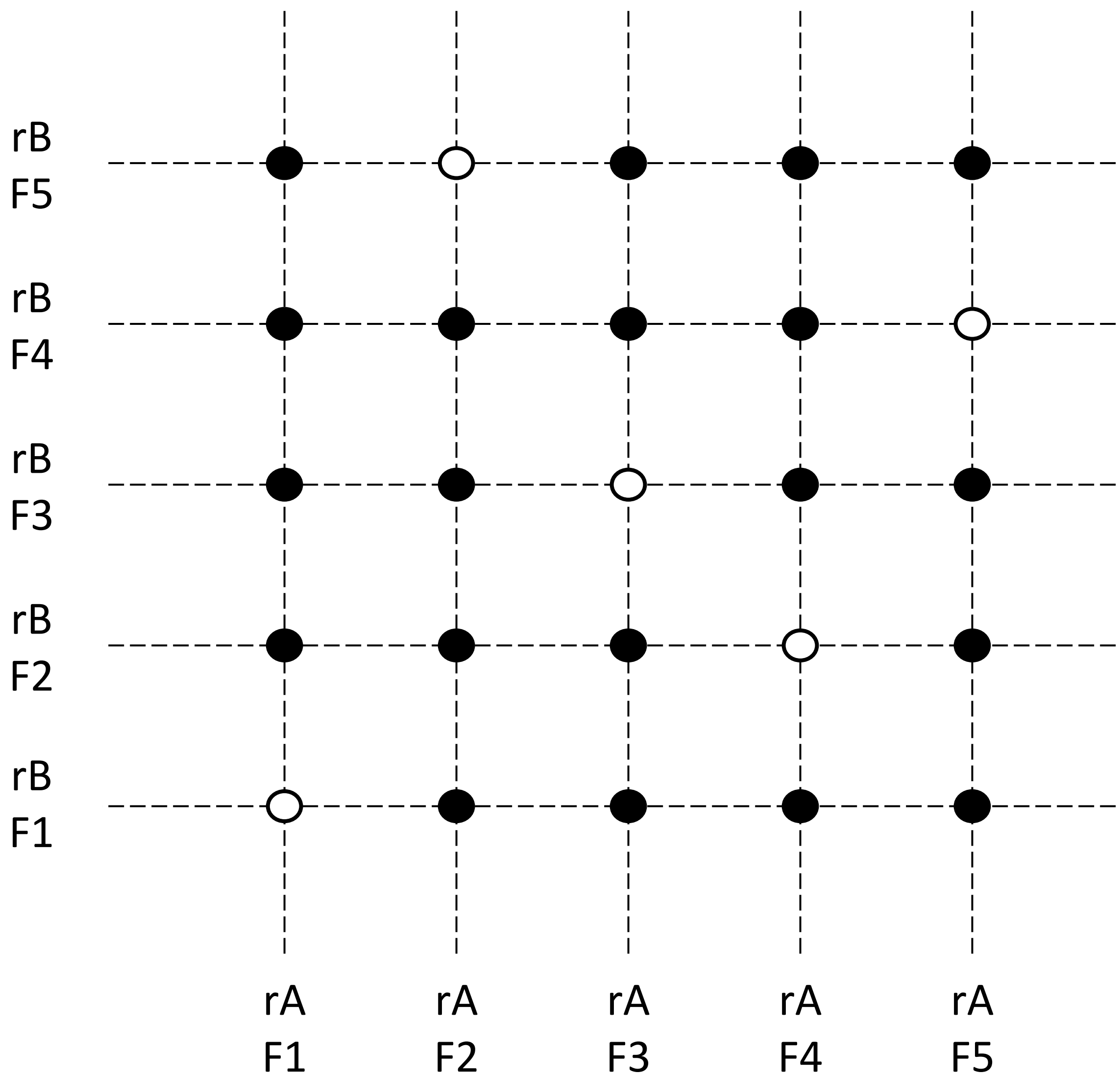
- There are only 2 functional requirements: rA and rB
- There are 5 alternative ways to satisfy rA, called rAF1 to rAF5
- There are 5 alternative ways to satisfy rB, called rBF1 to rBF5

I'll give a simple visualization of this, on the next slides.

Step 1: Draw the space of potential specifications.



Step 2: Let full circles be specifications that satisfy minimal RE problem.

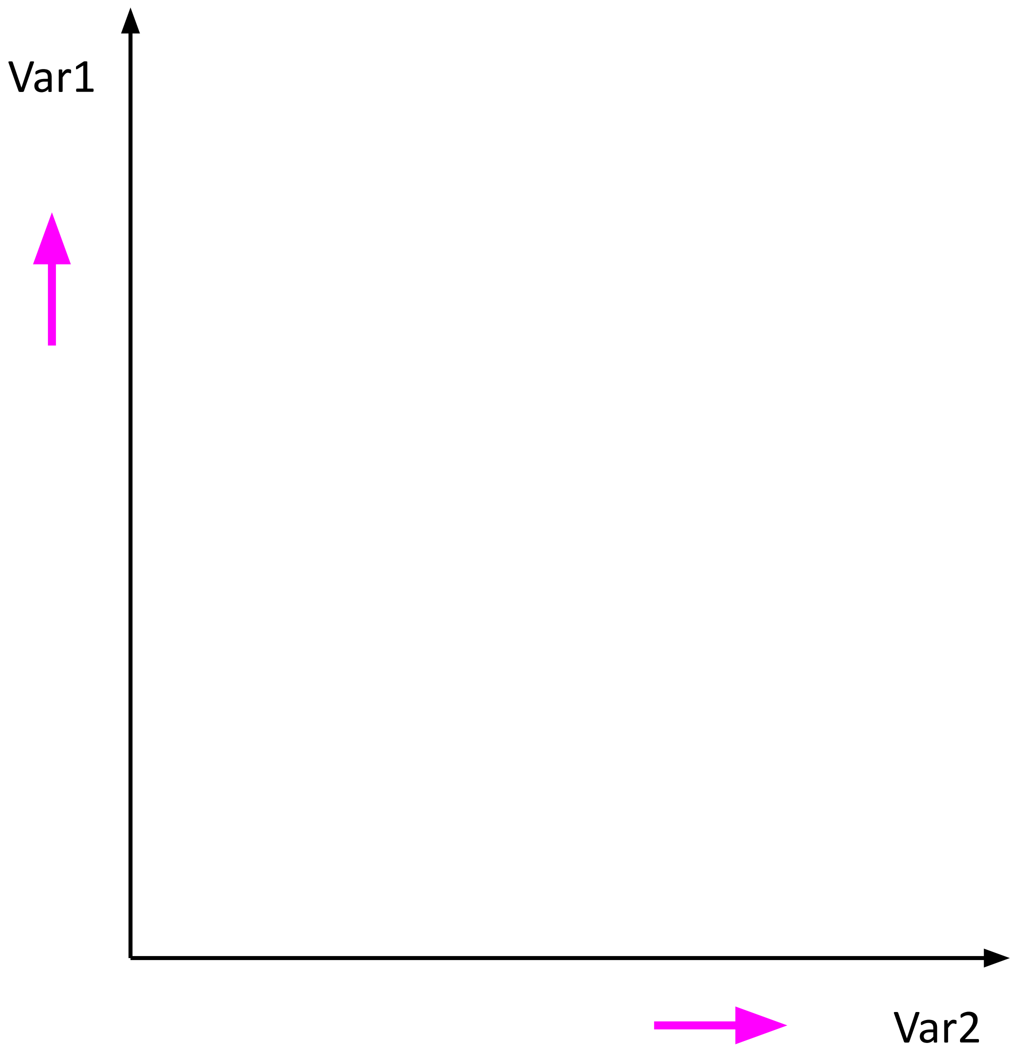


Step 3: Suppose that:

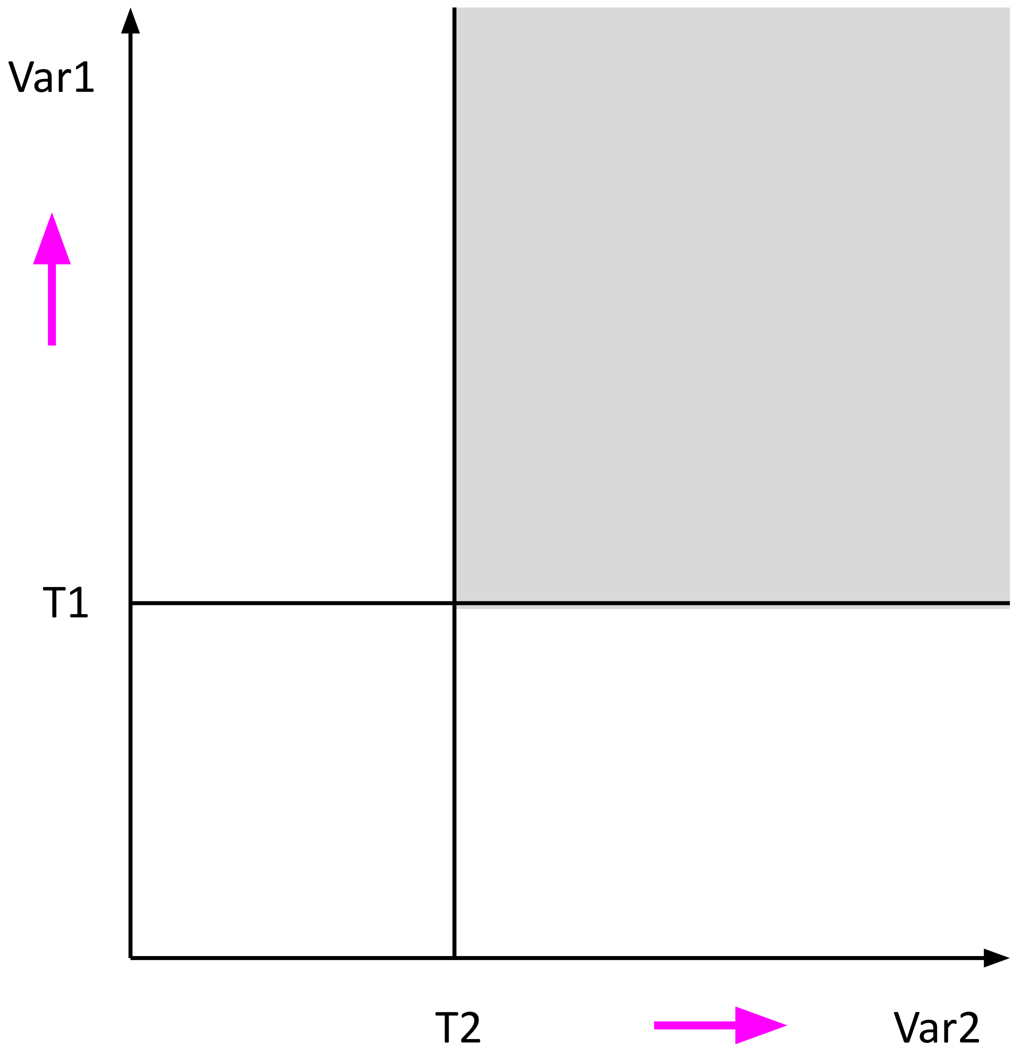
- we have two nonfunctional requirements,

- quantified by positive real-valued variables Var1 and Var2, and

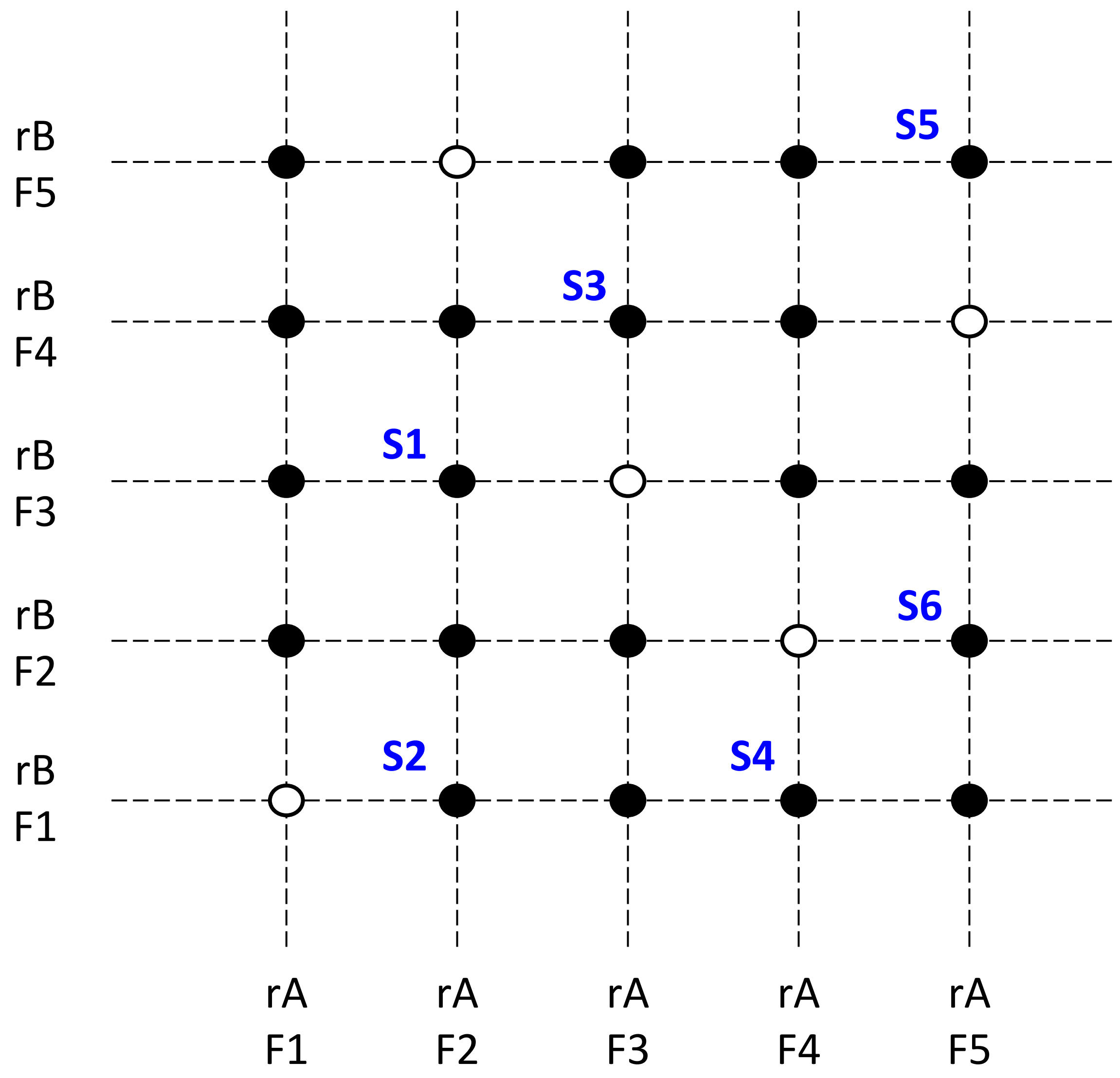
- we prefer higher than lower values of both variables



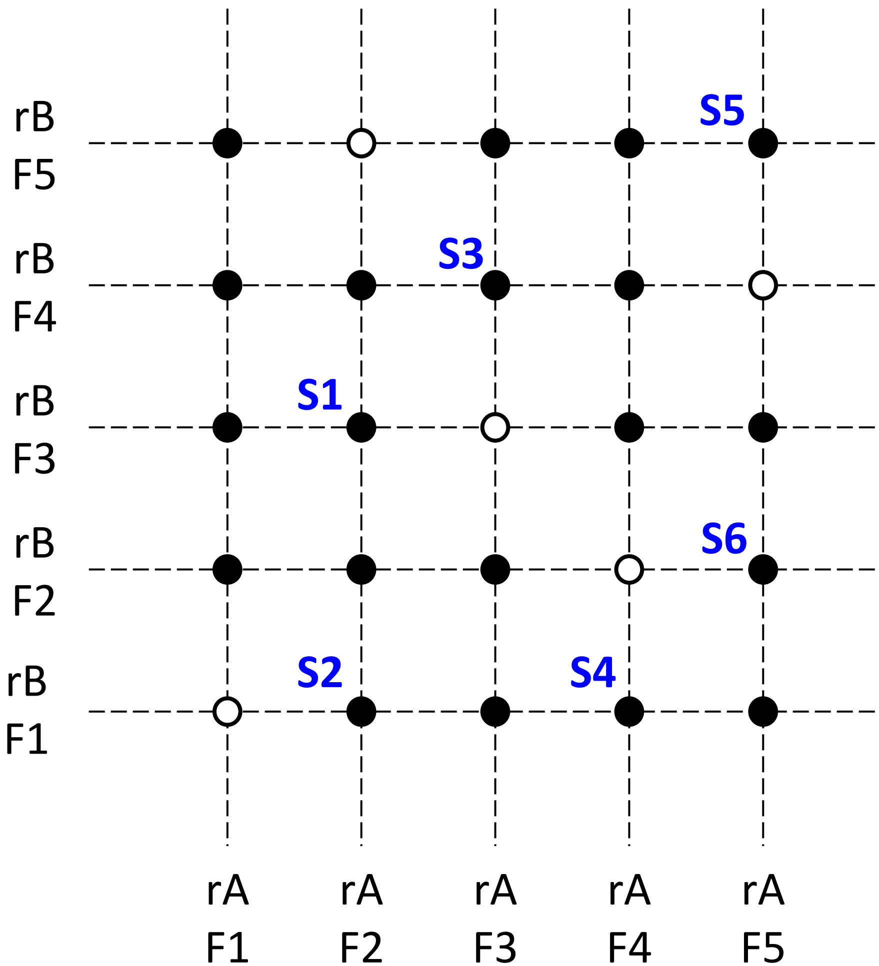
Step 4: We do NOT accept specifications with values below thresholds T1 and T2

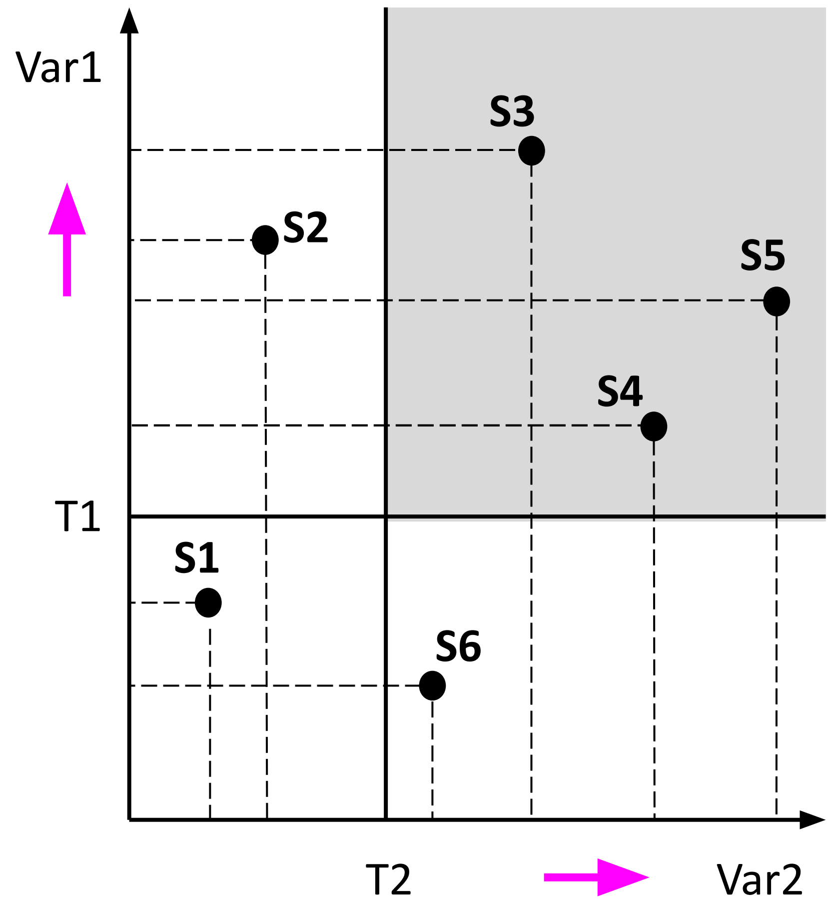


Step 5: Choose some of these specifications:



Step 6: Simulate chosen specifications, to get the levels to which they satisfy nonfunctional requirements.





Reminder - I claimed this:

In ASRE, we are NOT still trying to solve the minimal RE problem, BECAUSE:

ASRE specification is not the minimal RE specification.

I'll now give arguments why I believe the above is correct.

Argument 1:

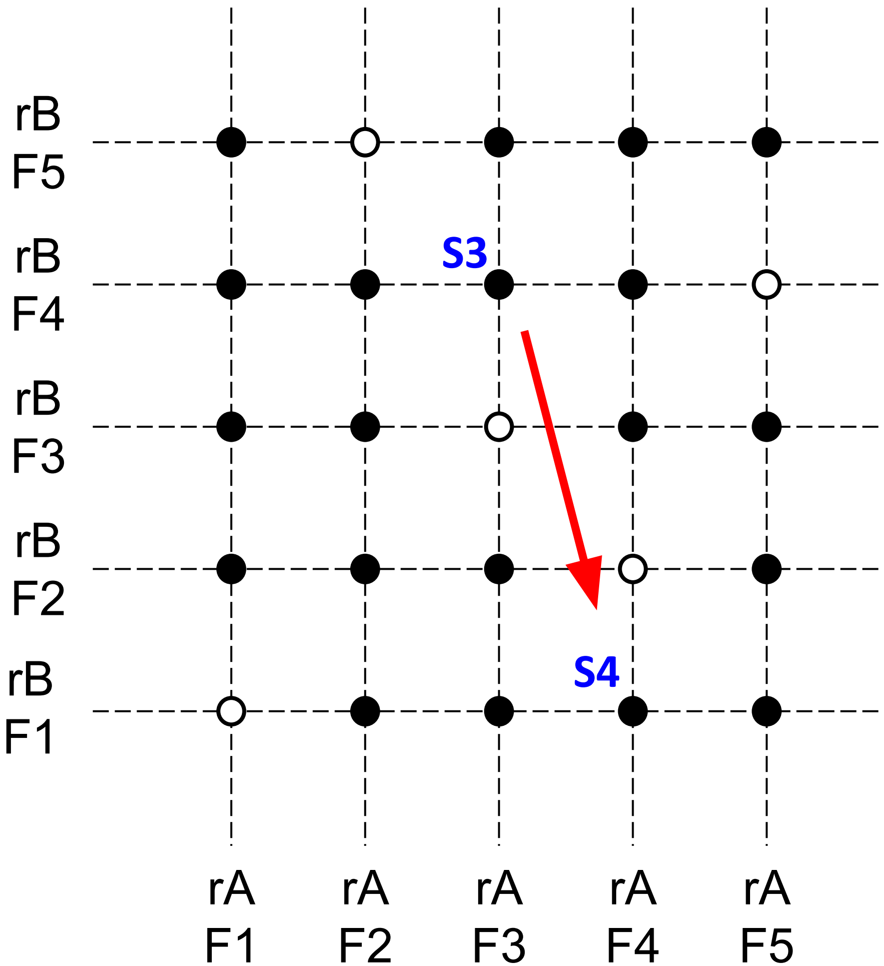
A system does not satisfy all nonfunctional requirements to the same level all the time.

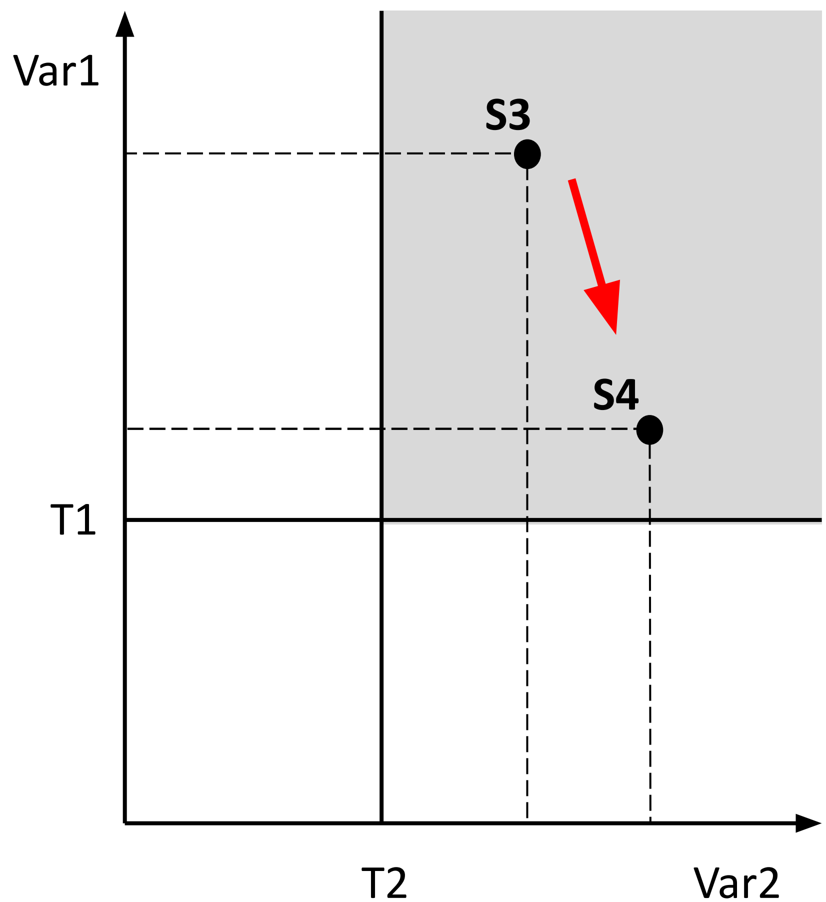
THEREFORE: An ASRE specification should define how the system should respond to such variation.

This is done via:
- Monitoring
- Feedback loops
- Evolution requirements

Argument 1 illustration

Assume that rBF4 fails. System switches from S3 to S4.





Can't we put monitoring and feedback loops in the specification that satisfies the minimal RE problem?

No, because:

S would have to include both S3 and S4 from the illustration.

BUT, S has to be consistent according to the minimal RE problem, yet S3 and S4 in the illustration need not be.

Argument 2:

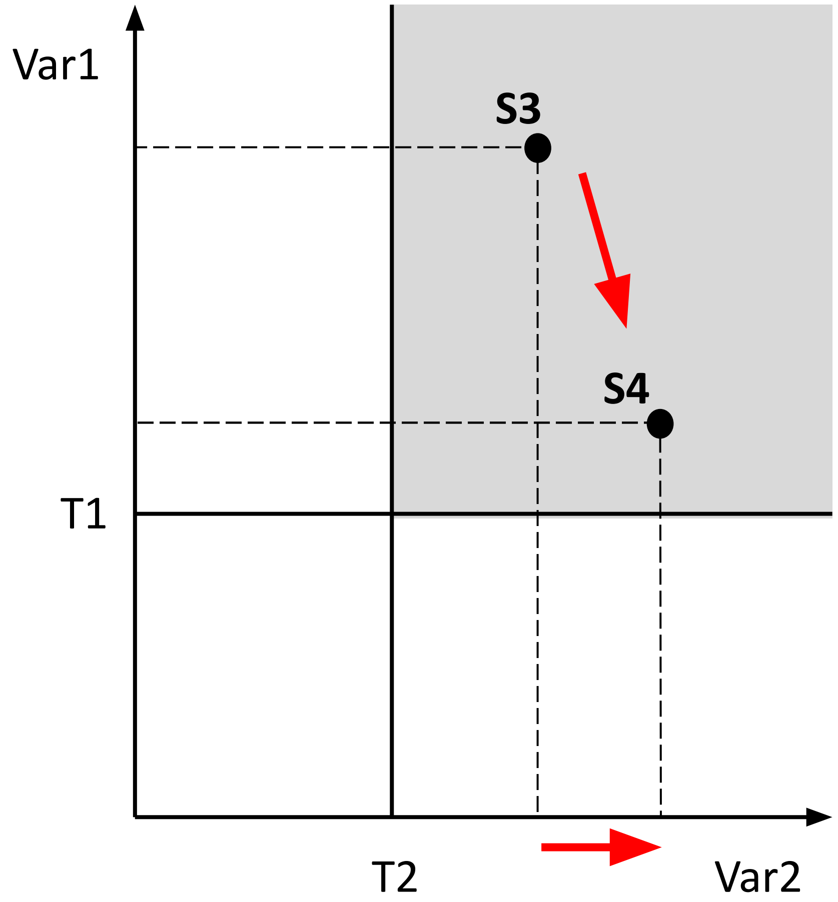
Over time, we normally want to optimize the level to which the system satisfies **both relaxed and nonfunctional requirements**.

THEREFORE: An ASRE specification should define the optimal value(s) of these requirements, and how to get to them.

This is done via:
- Relaxation
- Evolution requirements

Argument 2 illustration

Assume that we want to increase the value of Var2.



Can't we have this notion of optimization in the minimal RE problem?

No, because:

Minimal RE problem is about what ONE specification should satisfy.

Optimization is about choosing the specification which best satisfies relaxed and nonfunctional requirements.

And searching for "more optimal" specifications over time.

Minimal ASRE problem (1/2)

Given requirements and domain knowledge, design the most preferred feasible requirements roadmap.

Requirements roadmap is the pair:

- **Set of Specifications**
- **Set of evolution requirements**

Minimal ASRE problem (2/2)

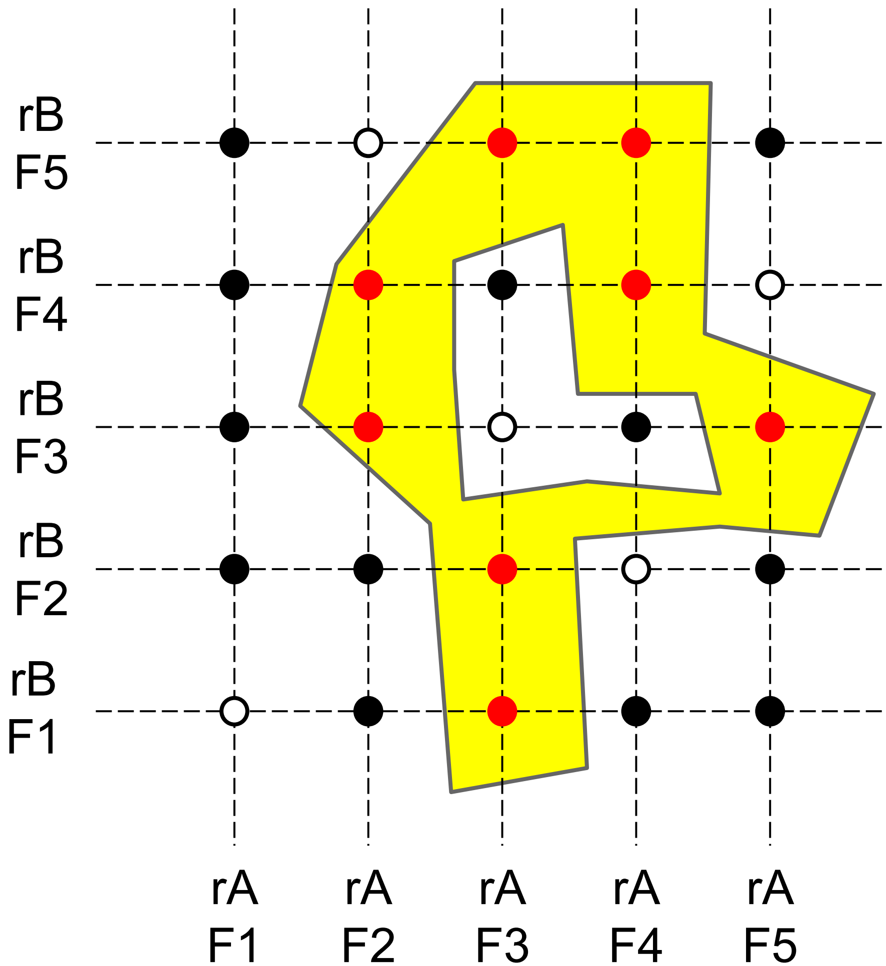
In a Requirements roadmap:

1. Each specification:
   a. Satisfies Minimal RE problem
   b. Achieves at least threshold levels of satisfaction for all relaxed and nonfunctional requirements

2. Each evolution requirement is an operator of the form <T, A, D>, where:
   a. T is a set of monitored requirements, in current specification
   b. A is a set of requirements that next specification satisfies
   c. D is a set of requirements from the previous specification, which must not be satisfied by the next specification.

Requirements roadmap illustration (1/2)

Set of specifications



Requirements roadmap illustration (2/2)

Set of evolution requirements