Università degli Studi di Trento
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dipartimento di Ingegneria e Scienza dell'Informazione

System Identification for Adaptive Software Systems: A Requirements Engineering Perspective

Vítor E. Silva Souza, Alexei Lapouchnian, John Mylopoulos

vitorsouza@disi.unitn.it
lapouchnian@disi.unitn.it
jm@disi.unitn.it

License to use, adapt and distribute

This material is available for any kind of use and can be derived and/or redistributed, as long as it uses an equivalent license and attributes credit to original authors.

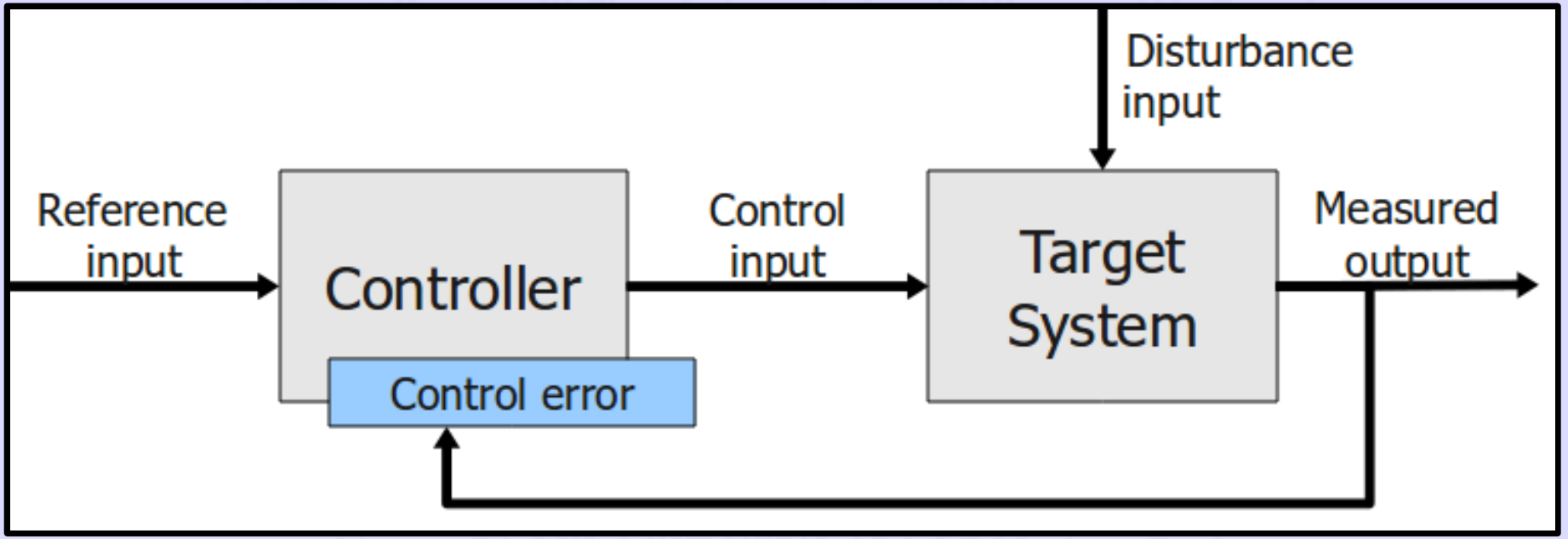
Attribution-Share Alike 3.0 Unported

http://creativecommons.org/licenses/by-sa/3.0/

You are free to copy, distribute, transmit and adapt this work under the following conditions: (a) You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work); (b) If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

Motivation

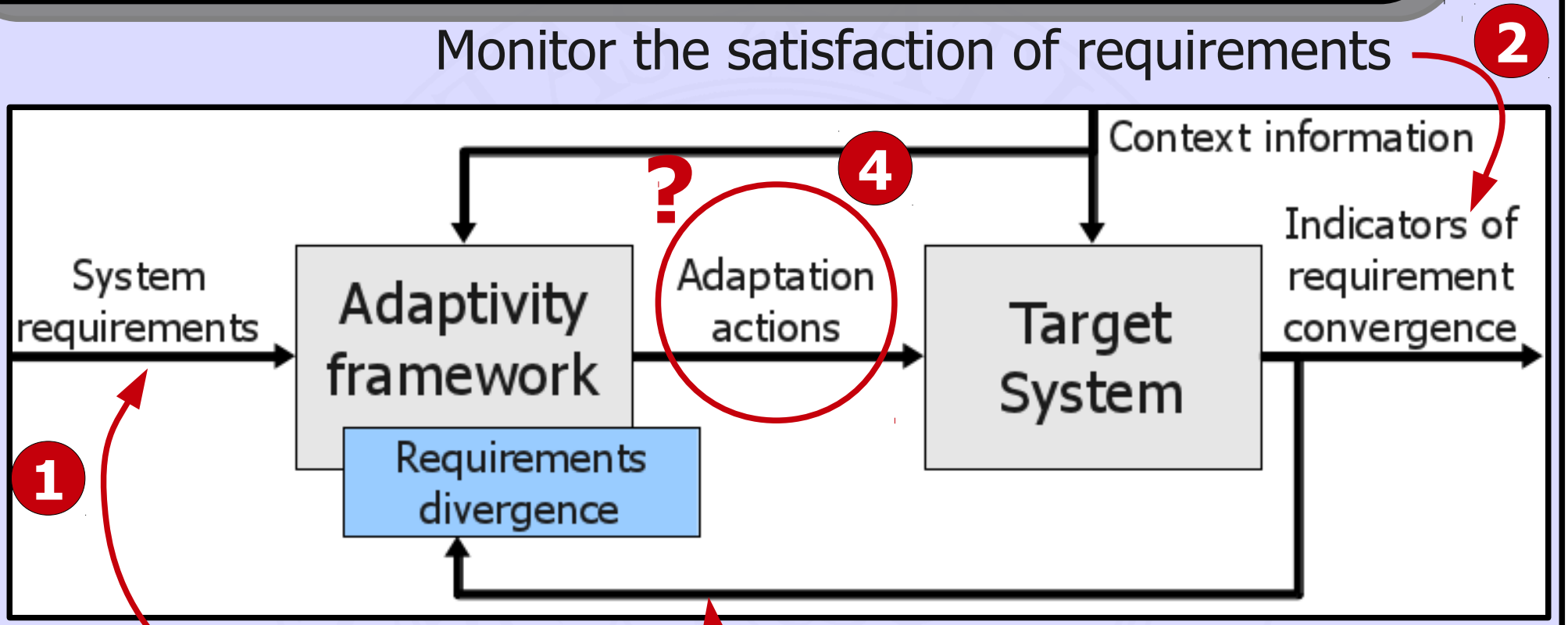
- Feedback loop architectures for adaptive systems [Andersson et al., 2009; Brun et al., 2009; Kephart & Chess, 2003].



Adapted from [Hellerstein et al., 2004]

System Identification: quantifying the effect of control inputs on measured outputs.

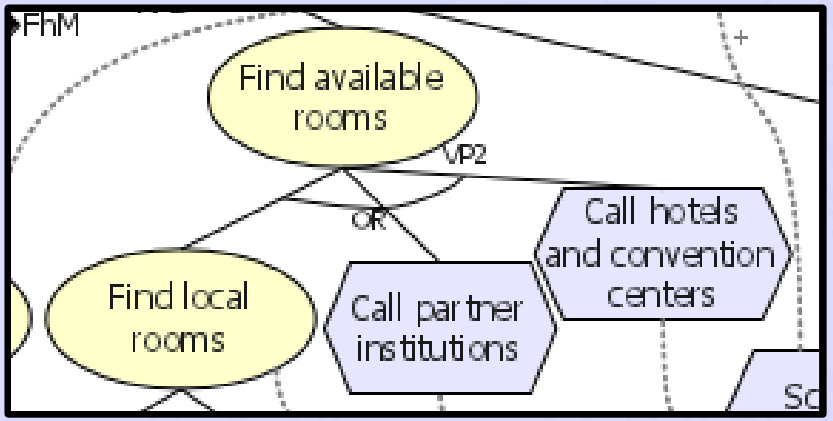
Feedback loops for adaptive systems



First principles (precise equations)	Initial model + free parameters	Observation and experimentation

Example / GORE perspective

1



Set the target for success rate of *Find available rooms* = 100%

2 At runtime, logs indicate that a user of the system was not able to find a room available;

3 The monitoring framework concludes there has been a requirements divergence;

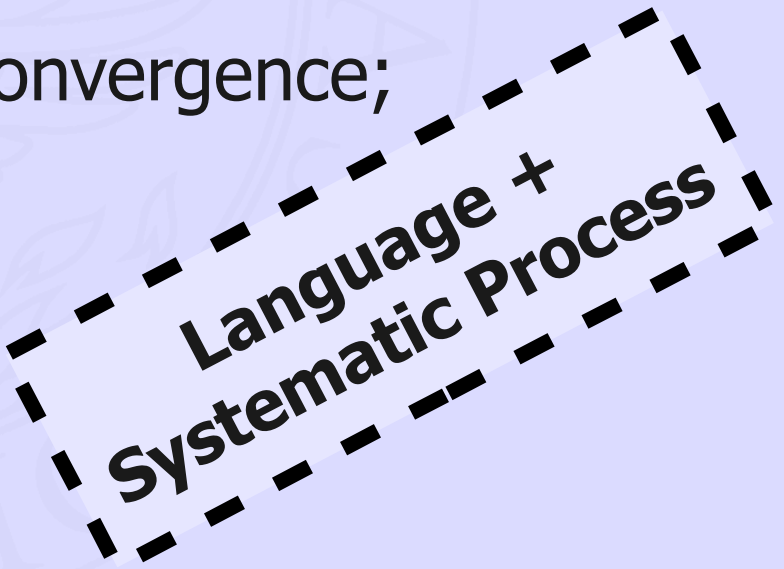
4 What can we do to adapt?

?
- Increase the number of rooms?
- Ask a partner institution for a room?
- Rent a room in a hotel or convention center?

Reconfiguration

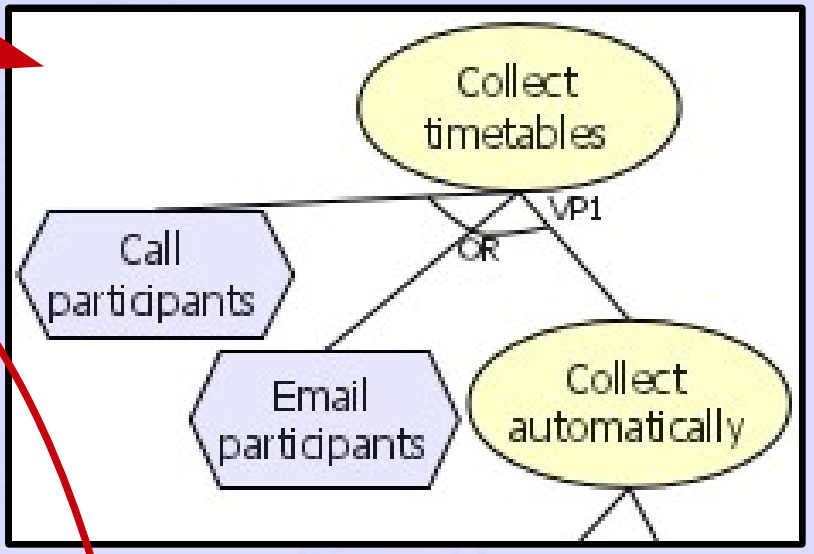
Problem and proposal

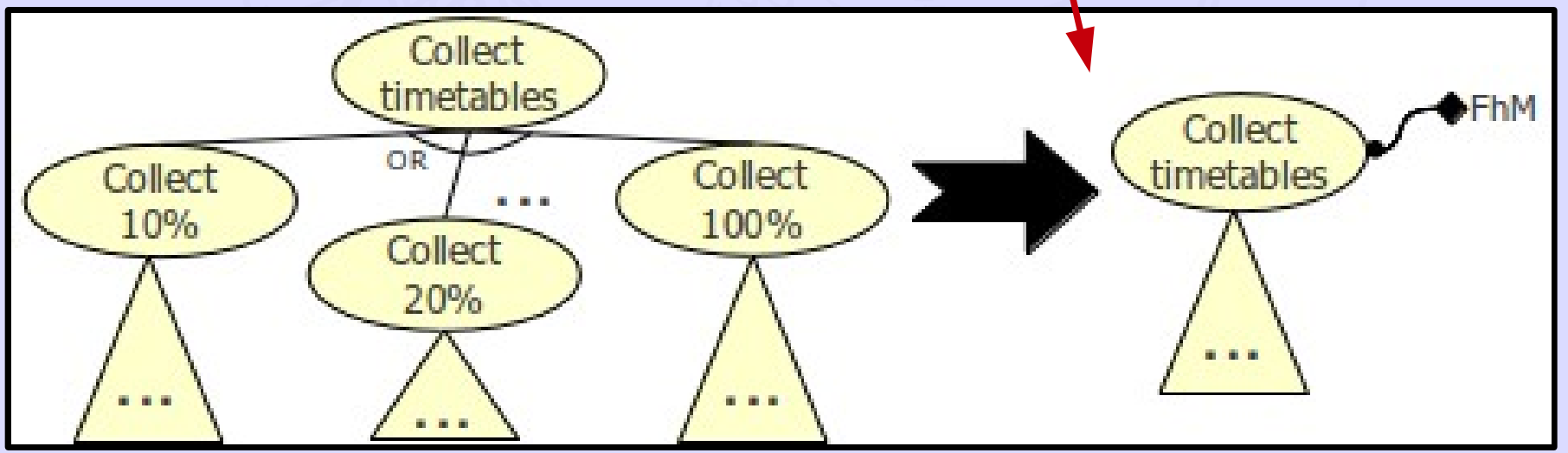
- Given a system that needs to be adaptive, perform system identification:
  - Systems-to-be or legacy systems.
- Augment the requirements (goal) model with:
  - System parameters;
  - Indicators of requirements convergence;
  - **Relations between them.**
- Relations are qualitative:
  - Ideas from Qualitative Reasoning [Kuipers, 1989].



Parameters: VPs and CVs

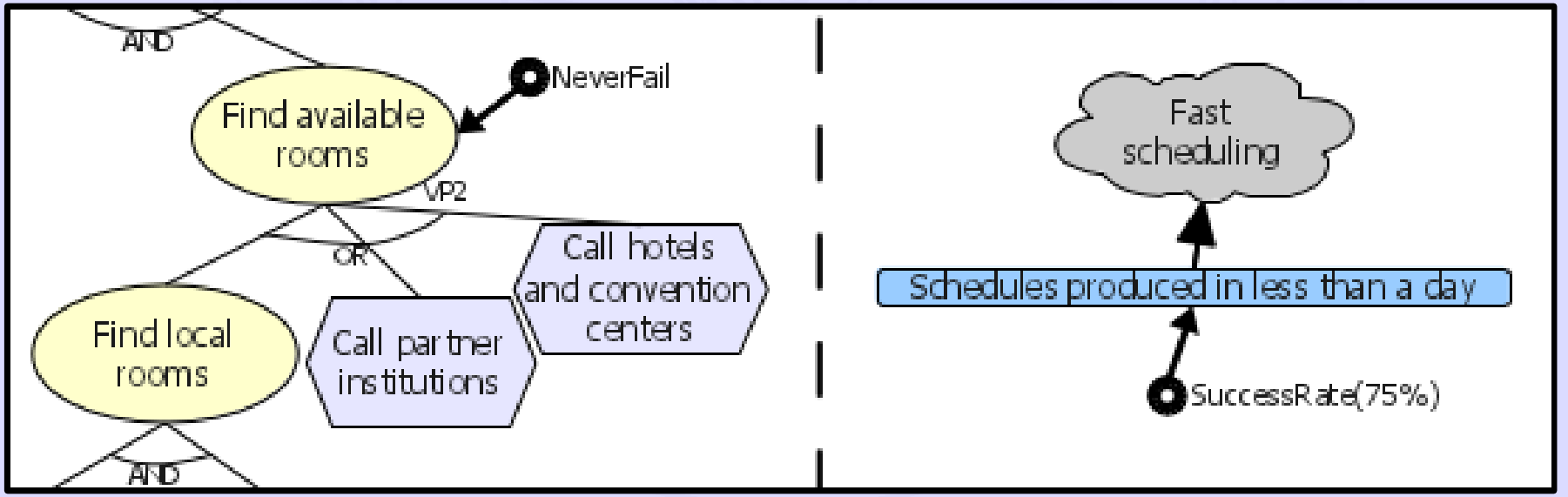
- Variation Points and Control Variables (numeric or enumerated);
- Been used for reconfiguration / reconciliation [Feather et al., 1998; Lapouchnian et al., 2007; Wang & Mylopoulos, 2009].



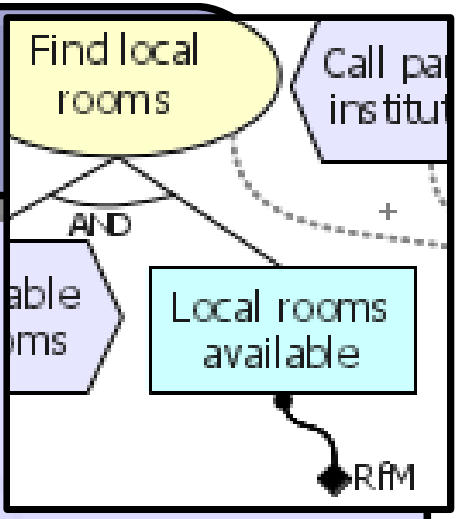


Indicators

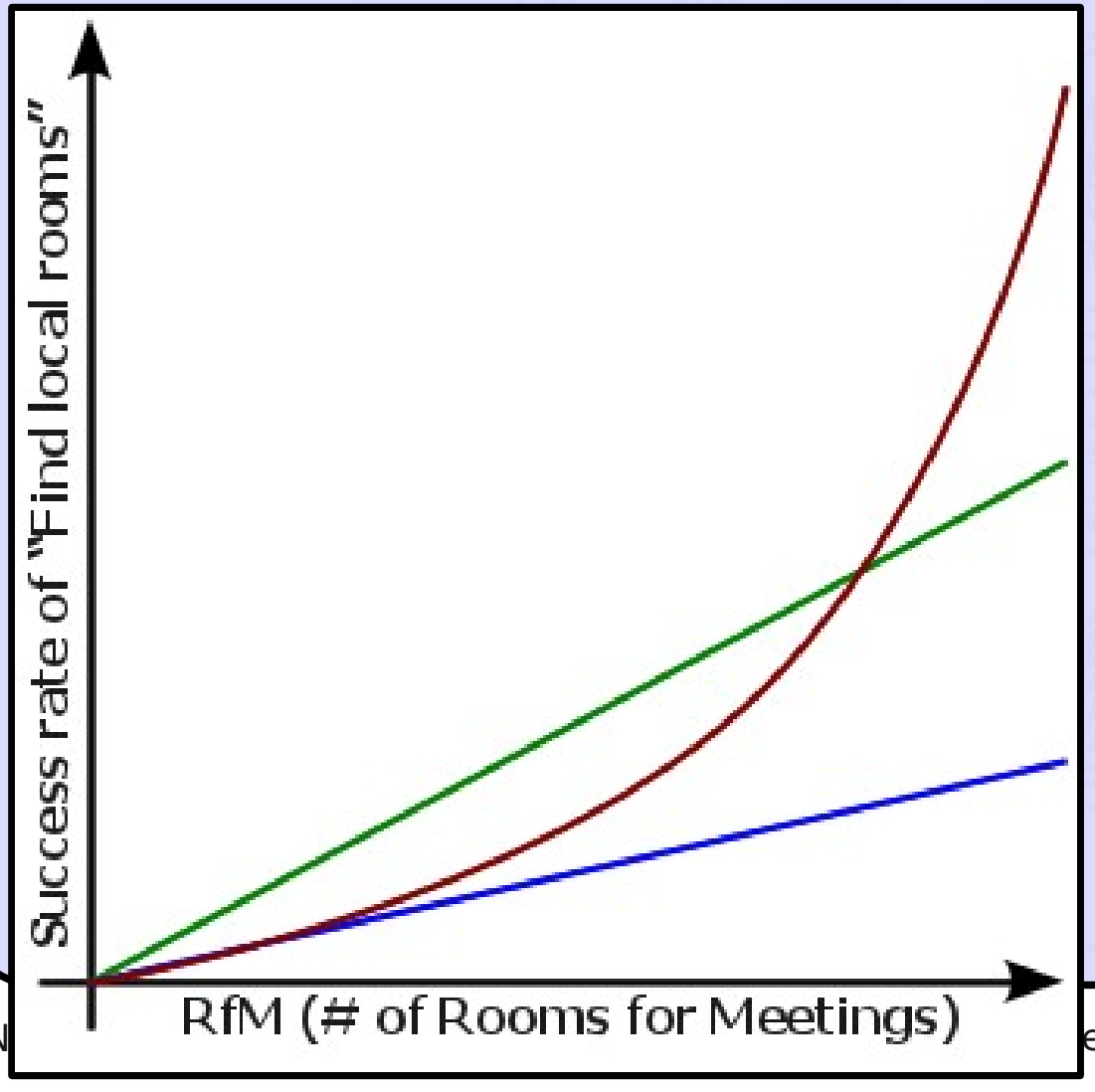
- Monitored, measurable value used as feedback:
  - Similar to gauge variables [van Lamsweerde, 2009].
- Focus on the important values the adaptive system should strive to achieve:
  - Awareness Requirements [Souza et al., 2011].



Indicator / parameter relation



- Control theory: difference equations;
- Indicators are affected by parameters.



Precisely how?

Which line best represents the relation?

We might not know!

Nature of the effect?

All of them have positive derivative.

Indicator / parameter relation

- Consider the simplification:

$$success\ rate\ of\ Find\ local\ room = f(RfM)$$

- Nature of this effect:

$$\frac{\Delta\langle success\ rate\ of\ Find\ local\ room\rangle}{\Delta RfM} > 0$$

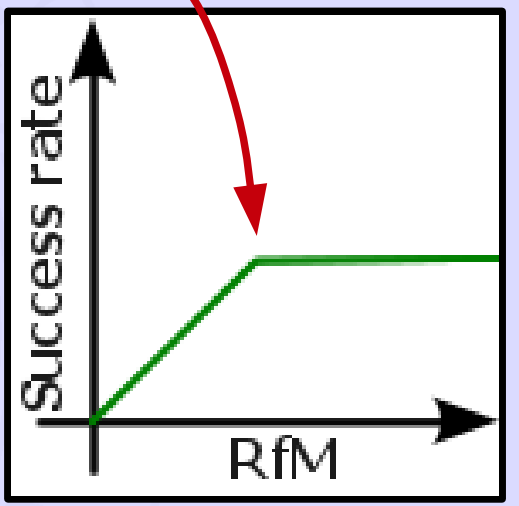
- General syntax in linearized notation:

Δ(indicator / param) <op> C

Landmarks, precision, extrapolations

- Boundaries for the effect:

  Δ(success rate... / RfM)[0, enoughRooms] > 0



- Still qualitative, but more precise:

  $\Delta(I / P_1) > 2$ $\Delta(I / P_2) > 5$

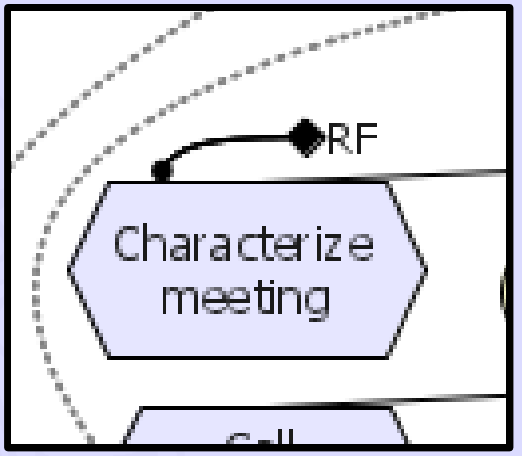
- Quantitative:

  $\Delta(I / P_1) = 2$ $\Delta(I / P_2) = 2P$

- Extrapolate a linearity assumption and the meaning of second derivative.

Enumerated parameters

- Examples:
  - Required Fields (RF): participant list only, short description required, full description required;
  - All variation points are enumerated.



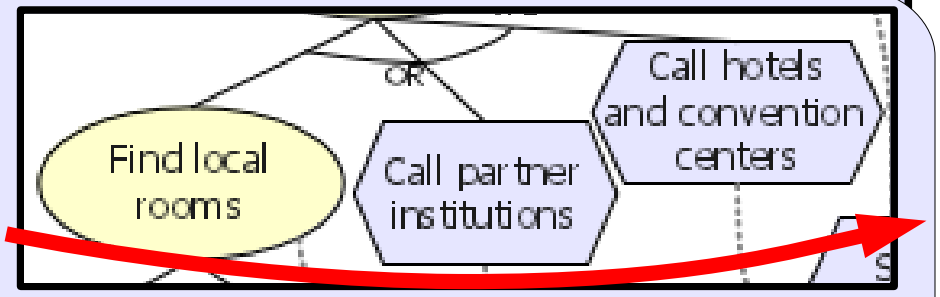
- Problem: sets are not total ordered. Solutions:

A $\Delta(I / P)\{\alpha_1 \rightarrow \beta_1, \ldots, \alpha_n \rightarrow \beta_n\} > 0$

B $\alpha_1 < \alpha_2 < \ldots < \alpha_n$

$\Delta(I / P) > 0$

C



Default VP order

Systematic process

- Input: requirements model G;
- Process:
  - 1 - Identify indicators I;
  - 2 - Identify parameters P;
  - 3 - Identify differential relations R(I, P);
  - 4 - Refine relations.
- Output: specification S = {G, I, P, R(I, P)}

Systematic process – example

1 - Indicator: AR5 = NeverFail(G-FindAvailRooms)

2 - Two parameters:

- RfM: the number of Rooms for Meetings;
- VP2: OR-refinement of goal *Find available rooms*

3 - Both parameters contribute positively:

- Δ(AR5 / RfM) > 0
- Δ(AR5 / VP2) > 0

4 - Equally effective and should not be combined:

- Δ(AR5 / RfM) = Δ(AR5 / VP2)
- Δ(AR5 / {RfM, VP2}) = Δ(AR5 / VP2)

Adaptivity example

- Specification from the example is provided;
- Monitoring [Souza et al., 2011] detects AR5 failure;
- Reconfiguration adaptation strategy analyzes the result of System Identification:

```
AwReq AR5 failed! To reconcile:
   - Current value of VP2 = local. Change to one
of: [partner, hotel].
   - Current value of RfM = 3. Increase it.
   - Note: VP2 and RfM should nto be changed in
combination.
```

Conclusion

- Information lacking for feedback loops:
  - How do system parameters affect indicators?
- Our contributions:
  - Extended goal model with concepts from Control Theory (parameters, indicators, relations);
  - A language to model parameter-indicator relations;
  - A systematic process for system identification.
- Our long-term goal:
  - Methods and tools for the design of feedback-loop based adaptive systems from requirements.

Future work

- What's the role of contexts in the produced specification?
- Techniques for estimating whether a particular change will match the desired targets;
- Chain effect: param → indicator 1 → indicator 2;
- More methods/concepts from Control Theory?
- Effect of this in traditional RE activities (stakeholder negotiation, requirements evolution, etc.);
- Full feedback loop (framework) + CASE tools;
- Further adaptation strategies.

Thank You! Questions?

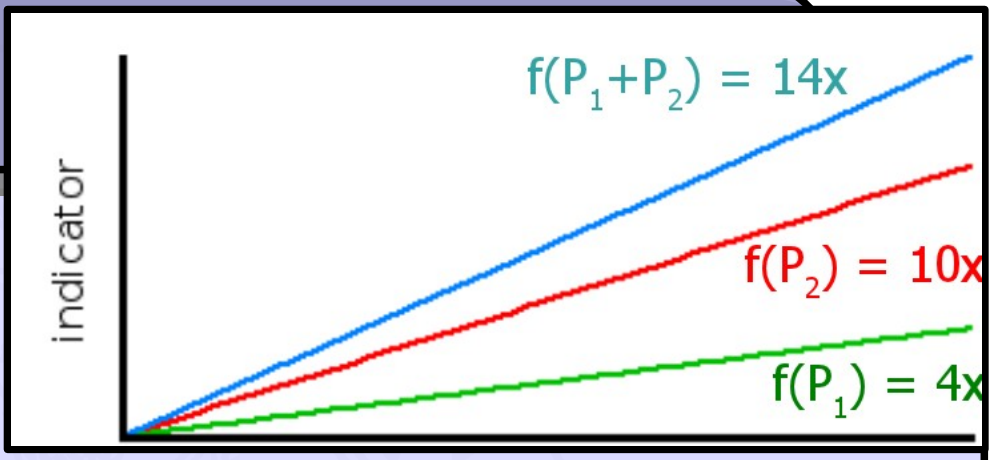
Acknowledgment:

The research reported in this presentation was partially funded by the ERC advanced grant 267856 "Lucretius: Foundations for Software Evolution", unfolding during the period of April 2011 - March 2016.

References

- J. Andersson, R. de Lemos, S. Malek, and D. Weyns. Modeling Dimensions of Self-Adaptive Software Systems, volume 5525/2009, pages 27–47, 2009.
- Y. Brun, G. Marzo Serugendo, C. Gacek, H. Giese, H. Kienle, M. Litoiu, H. Müller, M. Pezzè, and M. Shaw. Engineering Self-Adaptive Systems through Feedback Loops, volume 5525/2009, pages 48–70, 2009.
- M. S. Feather, S. Fickas, A. Van Lamsweerde, C. Ponsard. Reconciling System Requirements and Runtime Behavior. In IWSSD '98: 9th International Workshop on Software Specification and Design. Washington, USA, p. 50, 1998.
- J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury. Feedback Control of Computing Systems. John Wiley & Sons, 2004.
- B. Kuipers. Qualitative reasoning: Modeling and simulation with incomplete knowledge. Automatica 25(4), p. 571-585, 1989.
- A. Lapouchnian, Y. Yu, J. Mylopoulos. Requirements-Driven Design and Configuration Management of Business Processes. In BPM '07: 5th International Conference on Business Process Management, Brisbane, Australia, 2007.
- J. O. Kephart and D. M. Chess. The vision of autonomic computing. Computer, 36(1):41–50, 2003.
- V. E. S. Souza, A. Lapouchnian, W. N. Robinson, and J. Mylopoulos. Awareness Requirements for Adaptive Systems. In SEAMS '11: 6th International Symposium on Soft. Eng. for Adaptive and Self-Managing Systems. Honolulu, USA, 2011.
- A. Van Lamsweerde. Reasoning About Alternative Requirements Options, Conceptual Modeling: Foundations and Applications, ch. 20, p. 380-397, 2009.
- Y. Wang, J. Mylopoulos. Self-repair Through Reconfiguration: A Requirements Engineering Approach. In ASE '09: 24th IEEE/ACM International Conference on Automated Software Engineering, Auckland, New Zealand, 2009.

Extrapolations



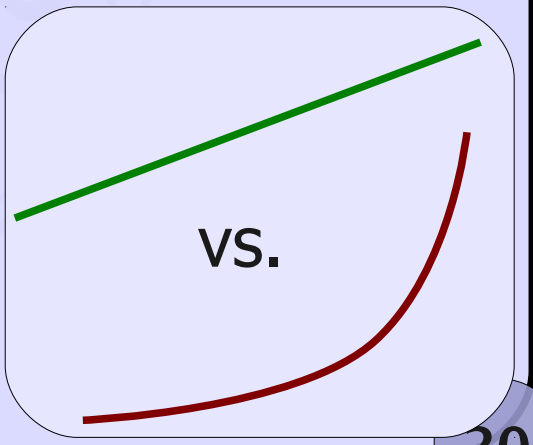
- Combining relations:
  - Linearity assumption;
  - Homogeneous impact is additive by default, unless otherwise specified:

$$\Delta(I / \{P_1, P_2, \ldots, P_n\}) = 0$$

- Second derivative:

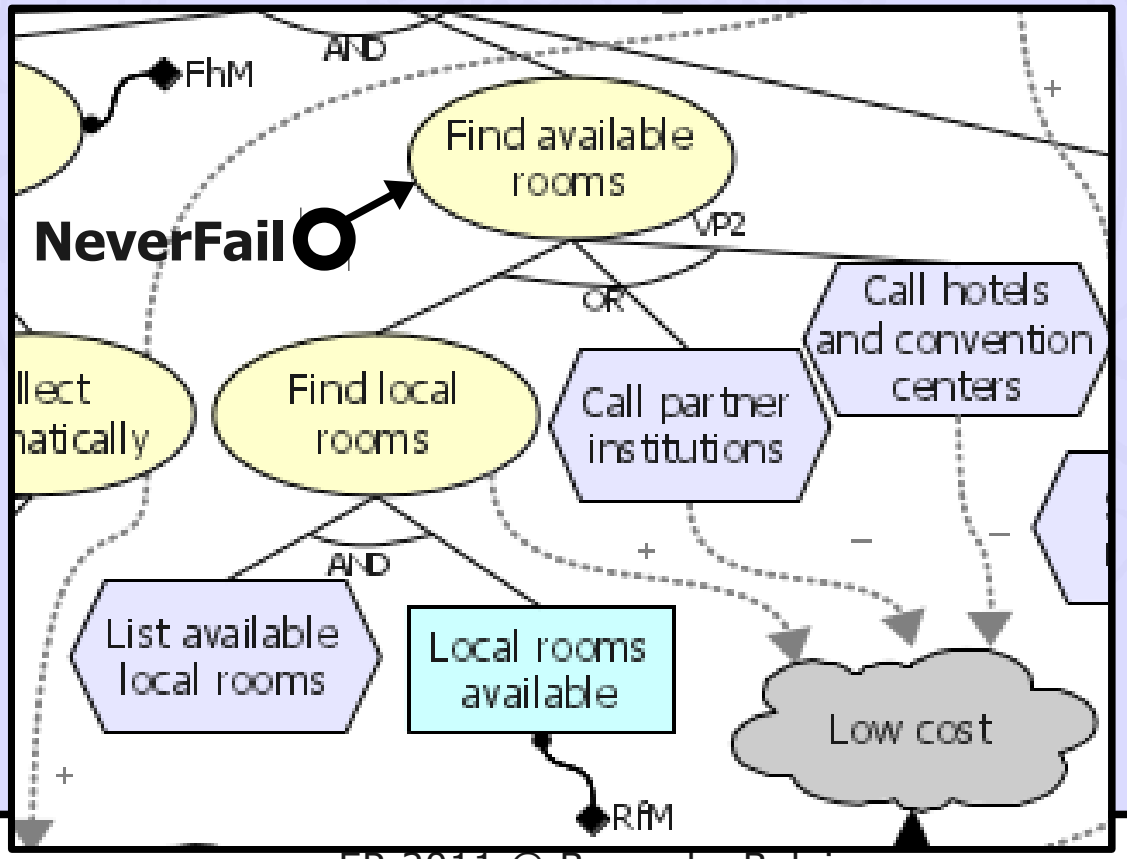
$$\Delta(I / P_1) > 0 \qquad \Delta^2(I / P_1) = 0$$

$$\Delta(I / P_2) > 0 \qquad \Delta^2(I / P_2) > 0$$



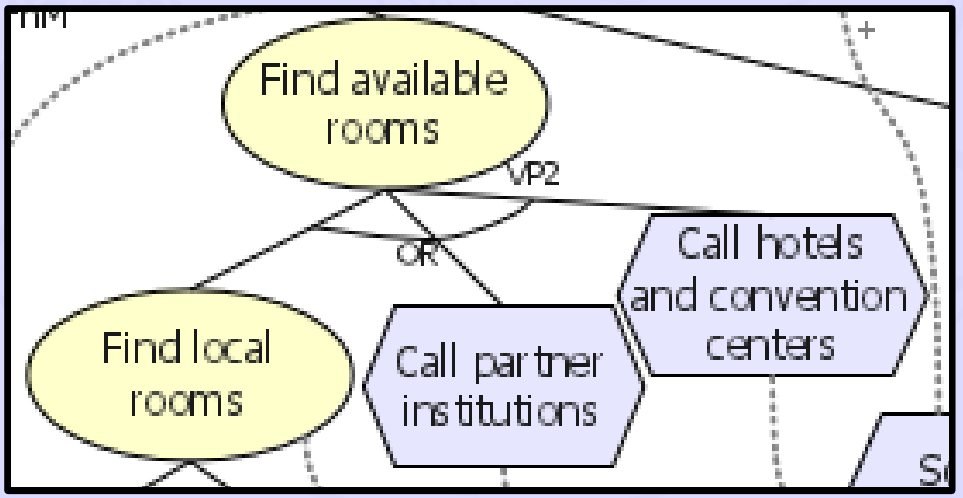
1 – Identify indicators

- Awareness Requirements [Souza et al., 2011];
- Example:
  - AR5 = NeverFail(G-FindAvailRooms)



2 – Identify parameters

- Identify variations in the goal model that can affect the indicators identified in step 1;
- Example:
  - RfM: the number of Rooms for Meetings;
  - VP2: OR-decomposition of the goal *Find available rooms*

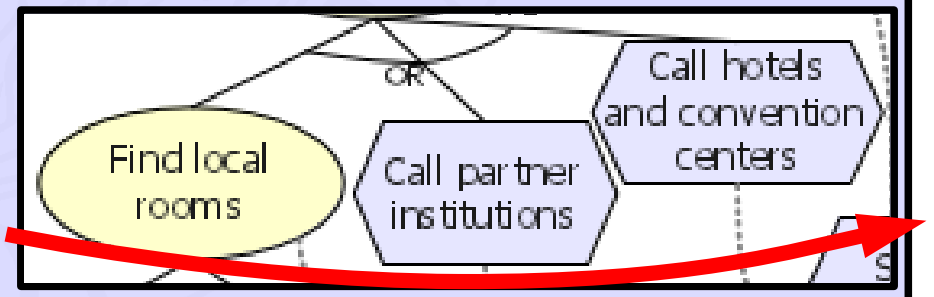


3 – Identify differential relations

- For each indicator, find parameters that affect it (alternatively, the other way around);
- Heuristics:
  - Analyze softgoal contributions;
  - Investigate the subtree of the node the indicator is associated with;
  - Consider non-functional requirements.
- Example:

$$\Delta(AR5 / RfM) > 0$$

$$\Delta(AR5 / VP2) > 0$$



Default VP order

4 – Refine relations

- Compare relations of the same indicator:
  - Is one relation "better" than the other?
  - Are homogeneous relations really additive?
- Example:

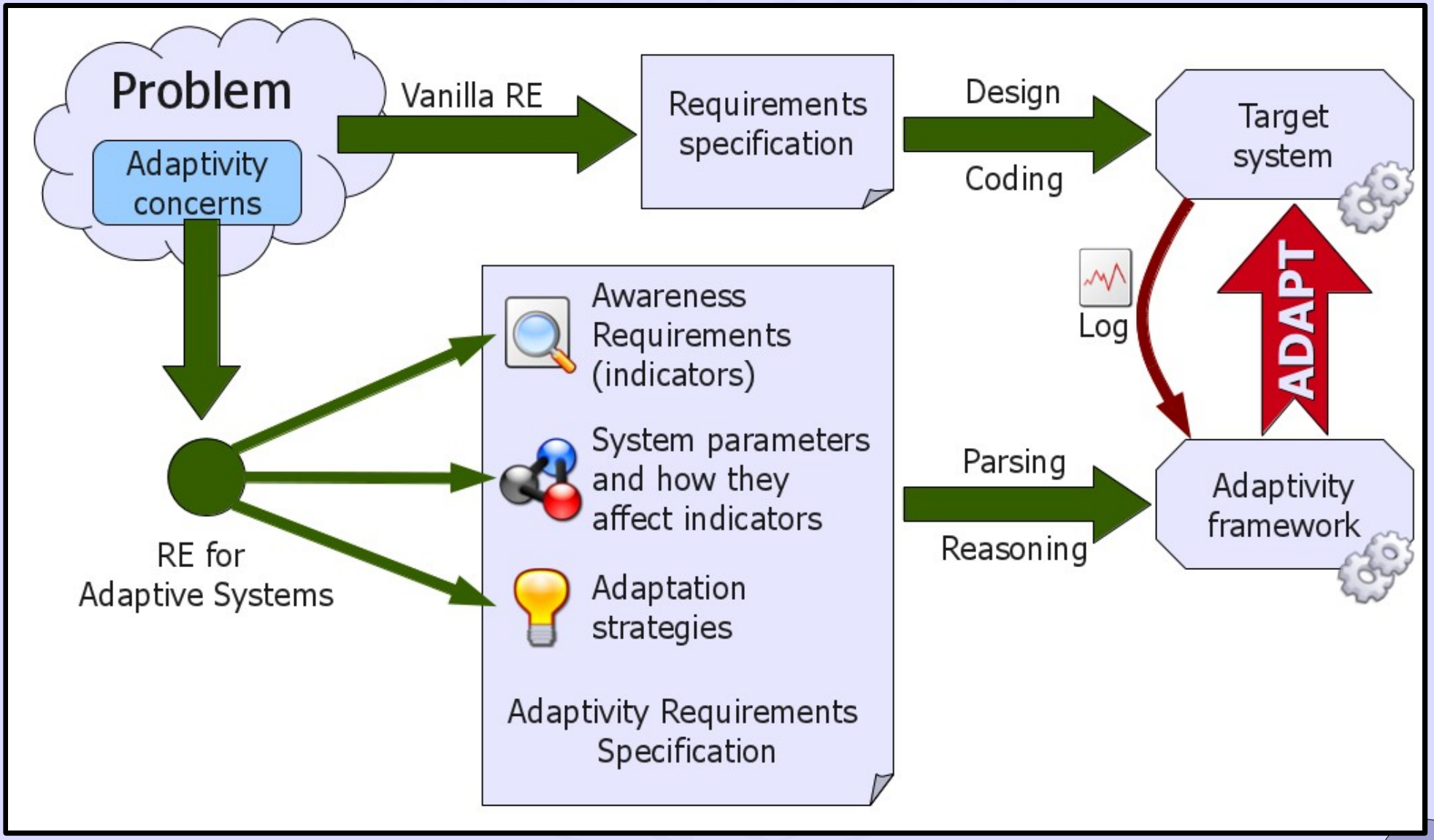
$$\Delta(\text{AR5} / \text{RfM}) = \Delta(\text{AR5} / \text{VP2})$$

RfM and VP2 are equally effective

$$\Delta(\text{AR5} / \{\text{RfM}, \text{VP2}\}) = \Delta(\text{AR5} / \text{VP2})$$

Changing from local rooms to something else and then increasing their number doesn't make sense.

The big picture



Related work

- Control Theory-based approaches for adaptivity [Heaven et al., 2009; Salehie & Tahvildari, 2011; Letier & Van Lamsweerde, 2004]:
  - Do not provide qualitative relations.
- Design-time trade-offs [Mylopoulos et al., 1999; Elahi & Yu, 2011; GRL]:
  - Can be adapted for runtime analysis;
  - Control parameters are more concise;
  - Differential relations are uniform and flexible.

Related work

- Automatic discovery of software tuning parameters [Brake et al., 2008]:
  - Our proposal takes a RE perspective, at a higher level of abstraction.
- Quantitative approaches [GRL; Cornford et al., 2006; Van Lamsweerde, 2009; Ma et al., 2009]:
  - Not suitable for domains with high uncertainty or incomplete knowledge of the system-to-be;
  - Our approach: start with minimum information, add more as they become available (at design-time or through run-time analysis).

Related work references

- GRL website: http://www.cs.toronto.edu/km/grl.
- N. Brake, J. R. Cordy, E. Dancy, M. Litiou, V. Popescu. Automating discovery of software tuning parameters. In SEAMS '08: International Workshop on Software Engineering for Adaptive and Self-managing Systems, 2008.
- S. L. Cornford, M. S. Feather, K. A. Hicks. DDP: a tool for life-cycle risk management. IEEE Aerospace and Electronic Systems Magazine 21(6), p. 13-22, 2006.
- G. Elahi, E. Yu. Requirements Trade-offs Analysis in the Absence of Quantitative Measures: A Heuristic Method. In SAC '11: 26th Symposium on Applied Computing, p. 2-4, 1996.
- W. Heaven, D. Sykes, J. Magee, J. Kramer. A Case Study in Goal-Driven Architectural Adaptation. In SEAMS '09: ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems, p. 109-127, 2009.
- A. Van Lamsweerde. Reasoning About Alternative Requirements Options, Conceptual Modeling: Foundations and Applications, ch. 20, p. 380-397, 2009.
- E. Letier, A. Van Lamsweerde. Reasoning about partial goal satisfaction for requirements and design engineering. In FSE '04: 12th ACM SIGSOFT International Symposium on Foundations of Software Engineering, p. 53-62, 2004.
- W. Ma, L. Liu, H. Xie, H. Zhang, J. Yin. Preference Model Driven Services Selection. In CAiSE '09: 21st International Conference on Advanced Information Systems Engineering, p. 216-230, 2009.
- J. Mylopoulos, L. Ching, E. Yu. From object-oriented to goal-oriented requirements analysis. Communications of the ACM 42(1), p. 31-37, 1999.
- M. Salehie, L. Tahvildari. Towards a Goal-Driven Approach to Action Selection in Self-Adaptive Software. Software Practice and Experience, 2011.

Università degli Studi di Trento
Facoltà di Scienze Matematiche, Fisiche e Naturali
Dipartimento di Ingegneria e Scienza dell'Informazione

System Identification for Adaptive Software Systems: A Requirements Engineering Perspective

Vítor E. Silva Souza, Alexei Lapouchnian, John Mylopoulos

vitorsouza@disi.unitn.it
lapouchnian@disi.unitn.it
jm@disi.unitn.it