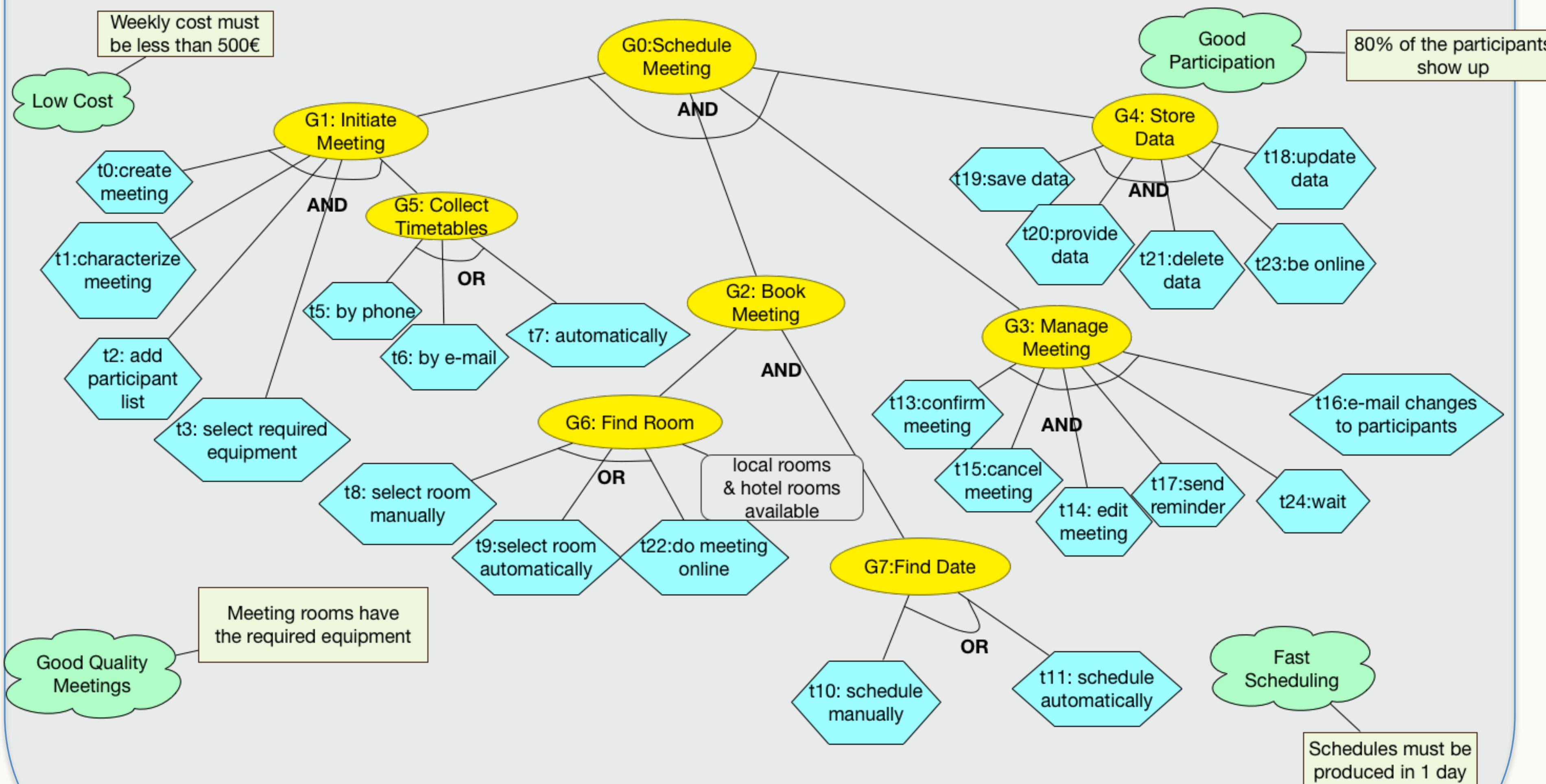
High Variability Models for Better Adaptive Systems

Konstantinos Angelopoulos
University of Trento

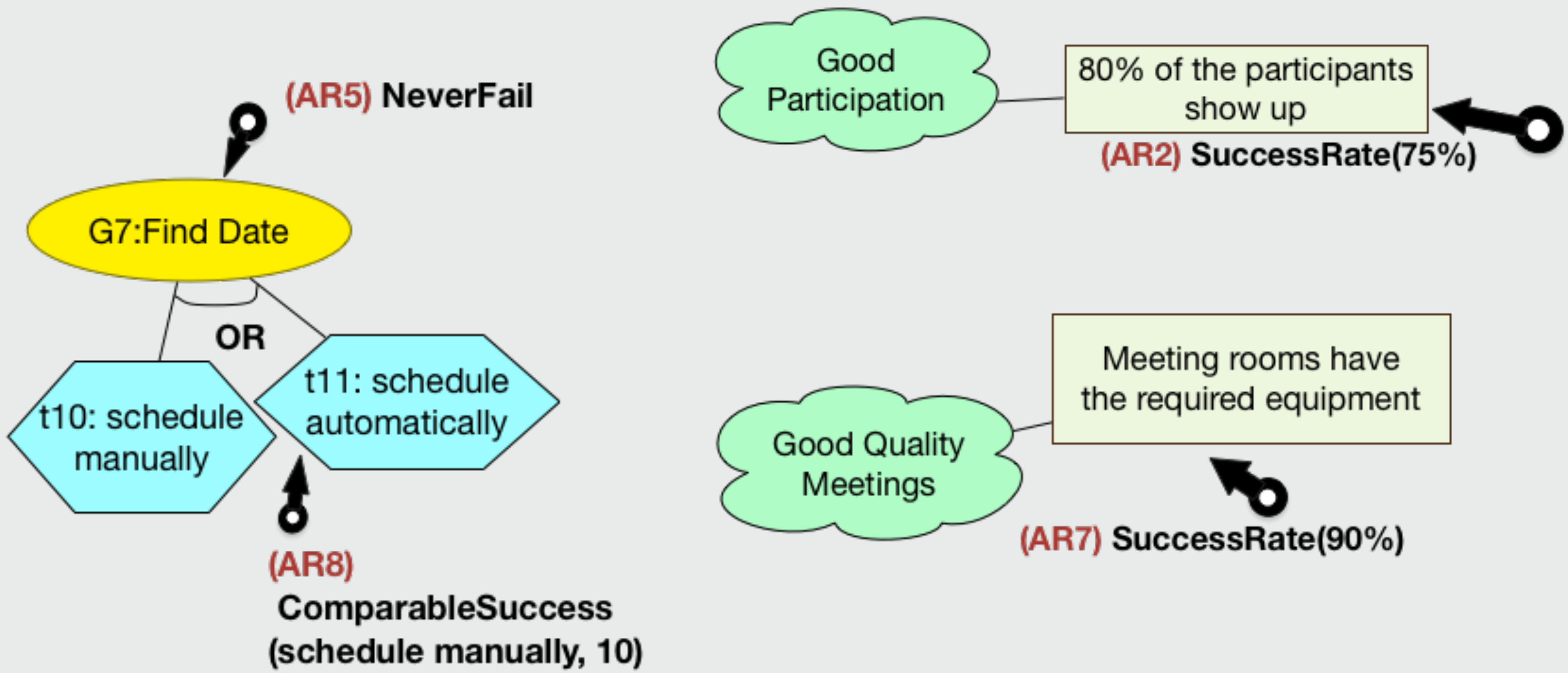
angelopoulos@disi.unitn.it

Baseline: Meeting-Scheduler



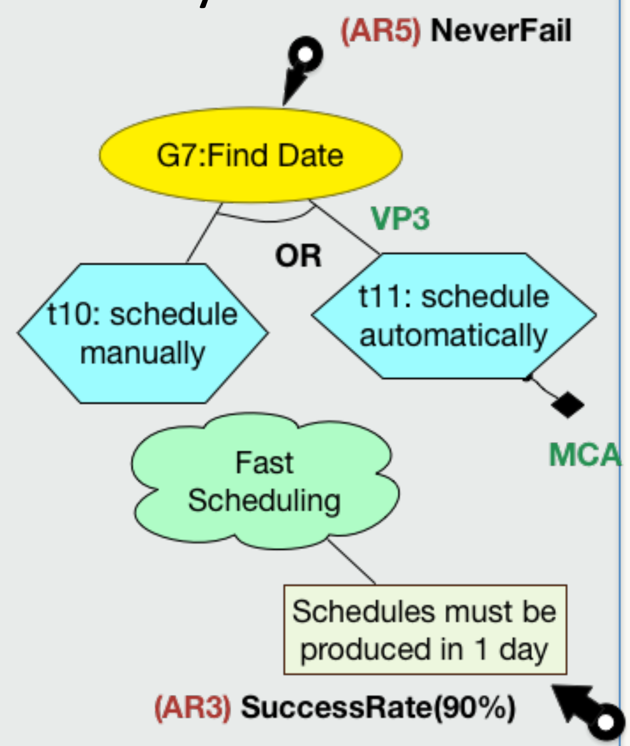
Baseline: Awareness Requirements

Awareness requirements: Define allowable thresholds on the success/failure of other requirements. Measured by variables called indicators.[Souza'11a]



Baseline: Control Parameters

System Identification: define the <u>parameters</u> of the system and the impact over indicators. [Souza'11b]



Maximum Conflicts Allowed (MCA)↑ then

Find Date (I5)↑, Good Participation ↓

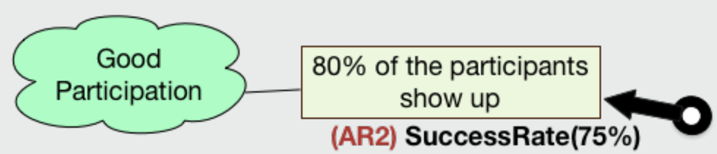
Differential relations:

Δ(I1/MCA)>0 , Δ(I2/MCA)<0

Switching from t10 to t11 (VP3) ↑ then

Fast Scheduling (I3) ↑ Δ(I3/VP3)[t10->t11]

Note: Differential relations are usually result of domain expertise and not physical laws.



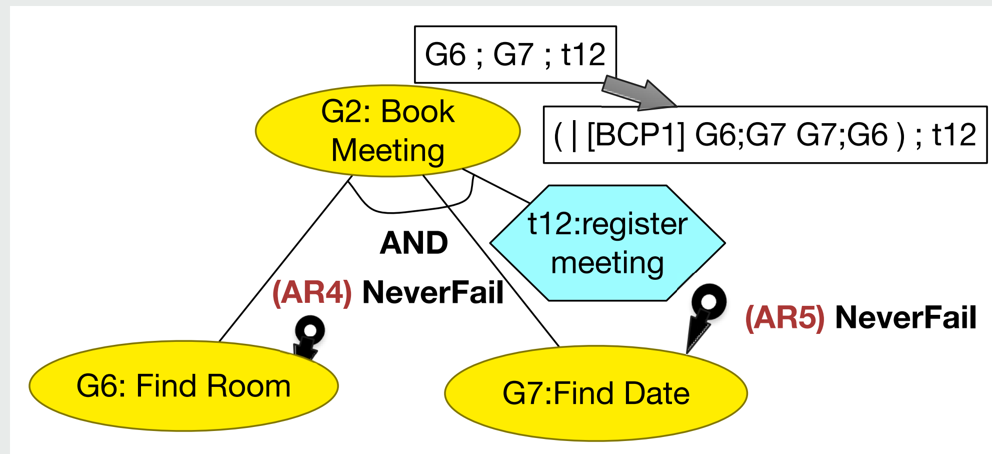
Is requirements variability all we have?

What about behavior and architecture?

Variability in Behavior

Meeting Scheduler:

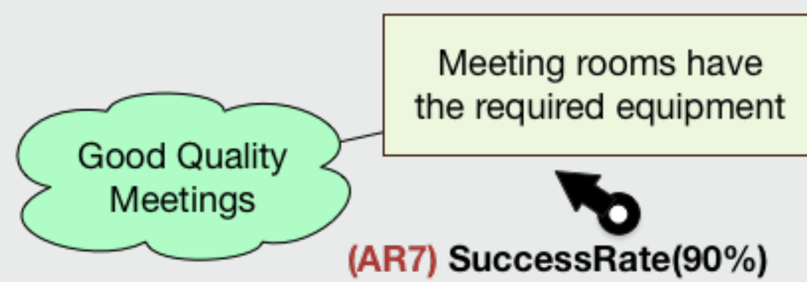
Both G6 and G7 must be fulfilled.

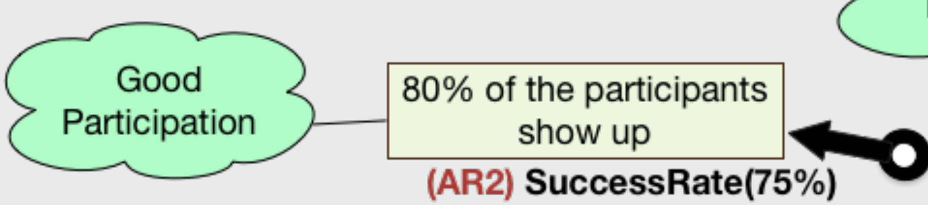


Differential relations with Behavioral Control Parameters [Ang'15a]:

Δ(I2 /BCP1)[G6; G7   G7; G6] > 0

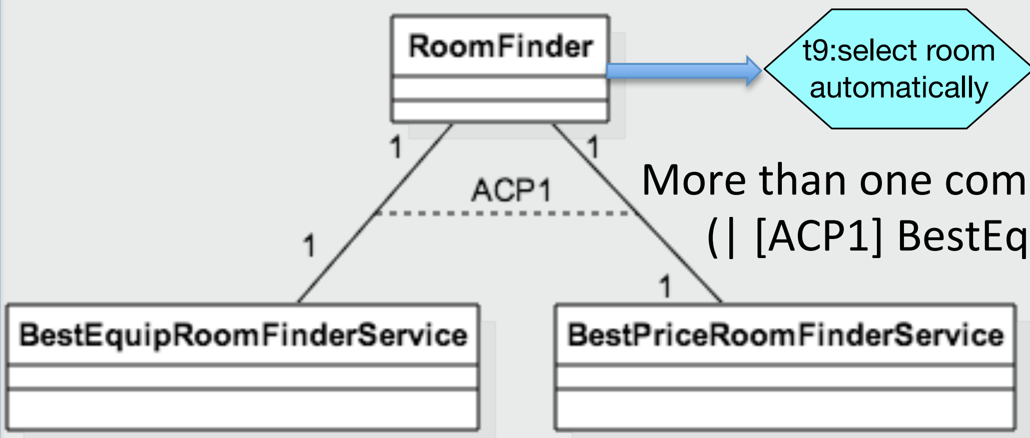
Δ(I7 /BCP1)[G6; G7   G7; G6] < 0



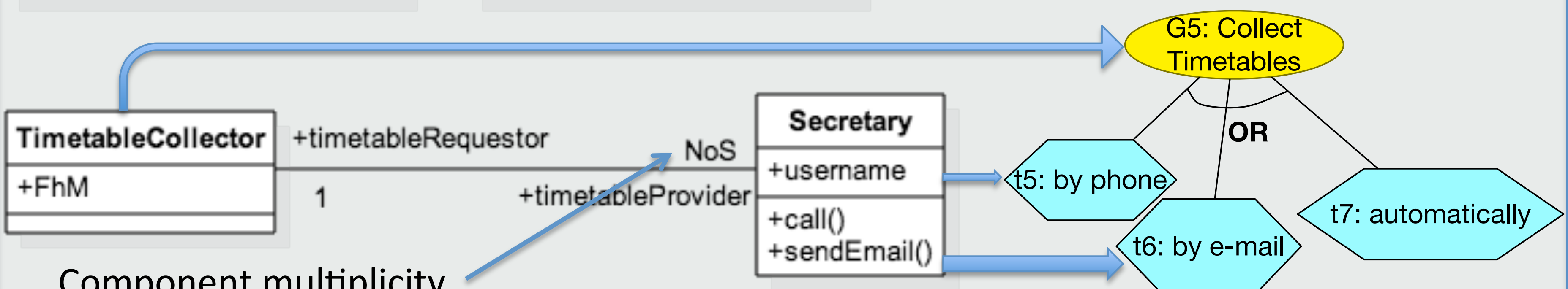


Variability in Architecture

Goals are fulfilled by components

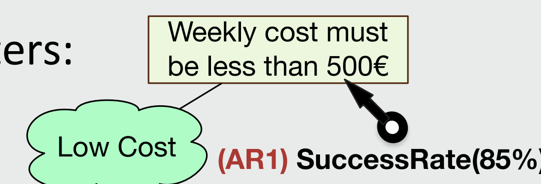


More than one components can fulfill the same requirement
(| [ACP1] BestEquip BestPrice BestEquip#BestPrice)



Component multiplicity

Differential Relations with Architectural Control Parameters:
$\Delta(I1/NoS) < 0$

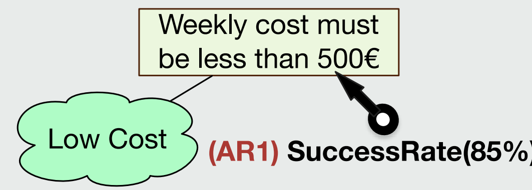


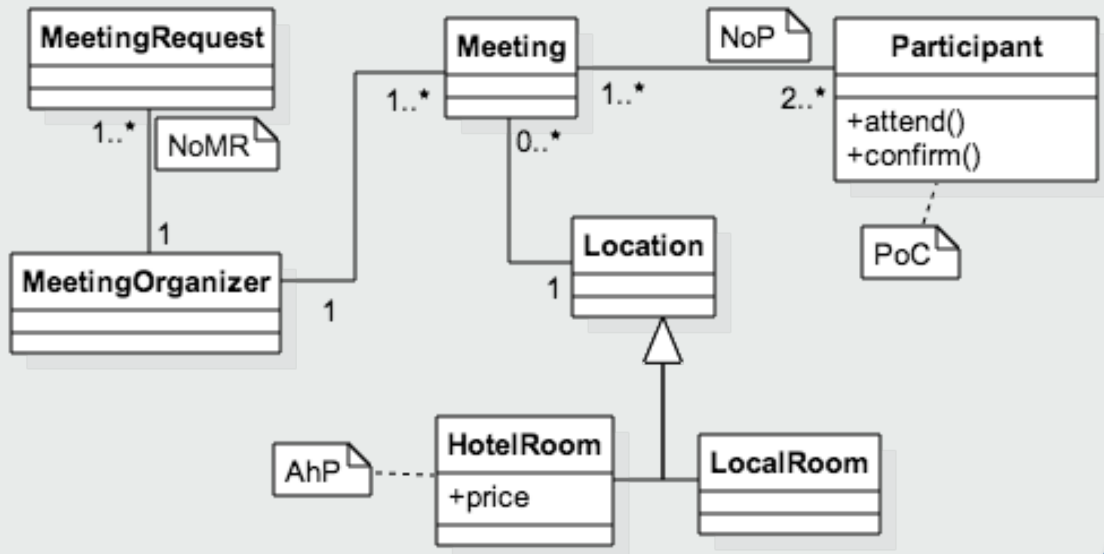
Variability in the Environment

In systems we can't control everything!

Environmental Parameters (EP):

- Percentage of consistency of the participants (PoC)
- Number of participants in the meetings (NoP)
- Number of meetings requested (NoMR)
- Average hotel prices (AhP)





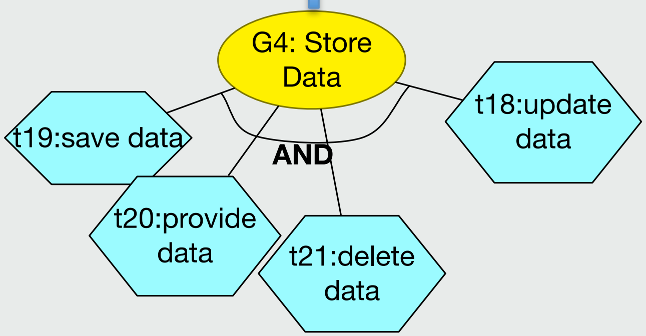
Differential relations with EP:

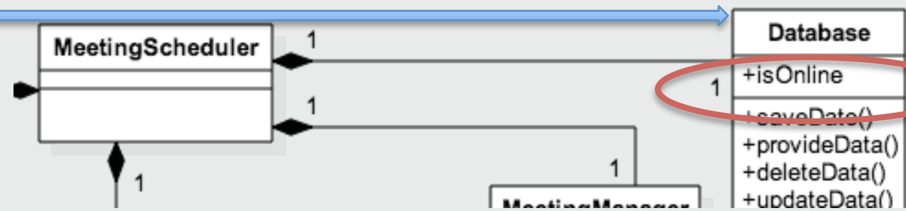
- Δ(I1/AhP)<0
- Δ(I1/NoMP)<0

Incremental Design: Three-Peaks Modeling

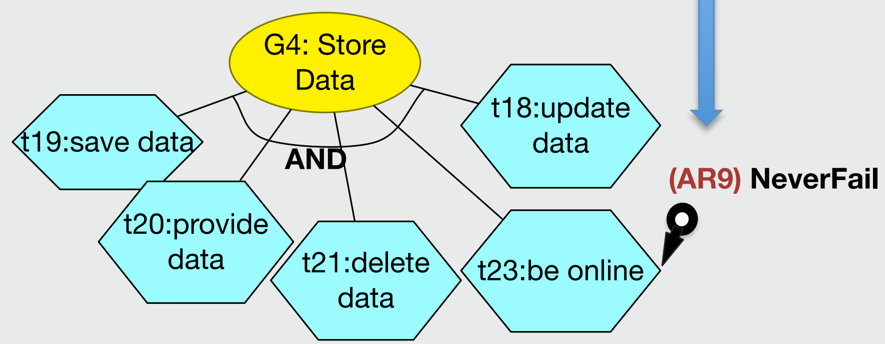
- Requirements and Architectures are intertwined [Nuseibeh'01]
- Along with Architecture we add Behavior [Angelopoulos'15a]
- More detailed the requirements become so is the system's architecture and behavior
- I'll know it when I see it (IKIWISI)
  - When new awareness requirements are specified, parameters must be designed to control them
  - When conflicts are present, new parameters must be introduced to handle them or refine the requirements
  - Architecture may impose its own constraints and therefore new requirements
- Three-Peaks Modeling: Define goals and behaviors, assign them to components in parallel and refine where there are risks of conflicts or new information appears

Incremental Design: Meeting Scheduler

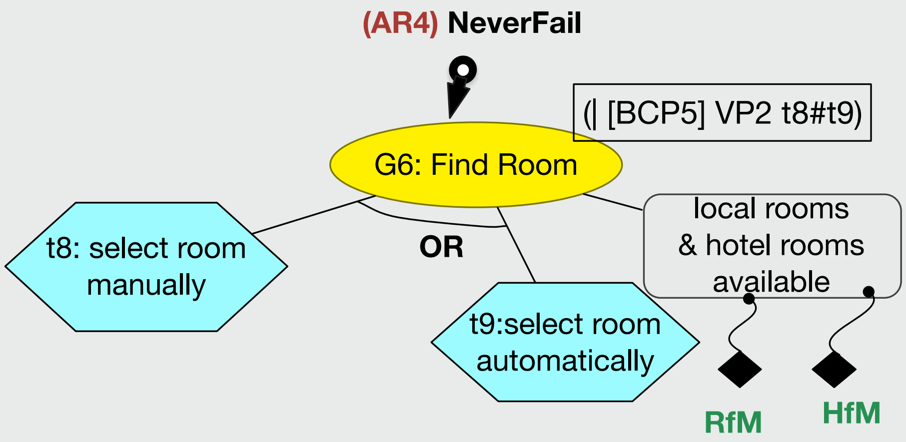




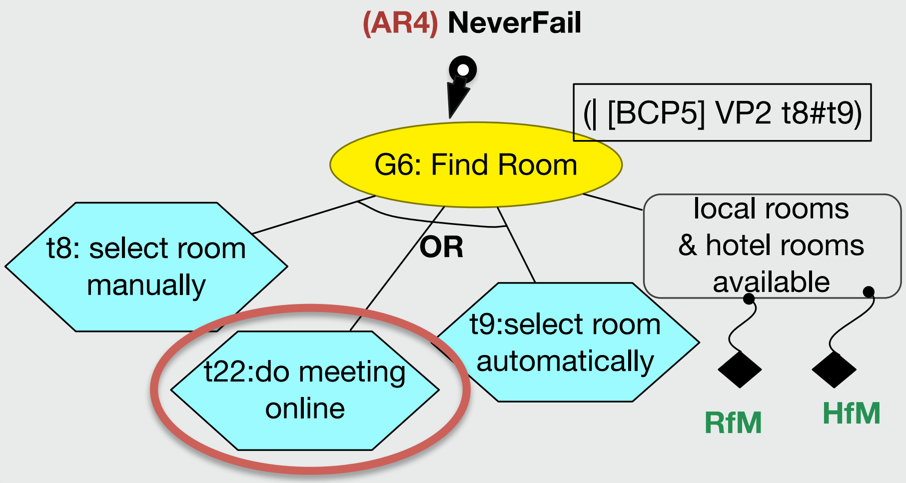
New architectural property leads to new requirements!



Incremental Design: Meeting Scheduler



Finding a room conflicts with many other requirements (e.g. Low Cost, Find Date)

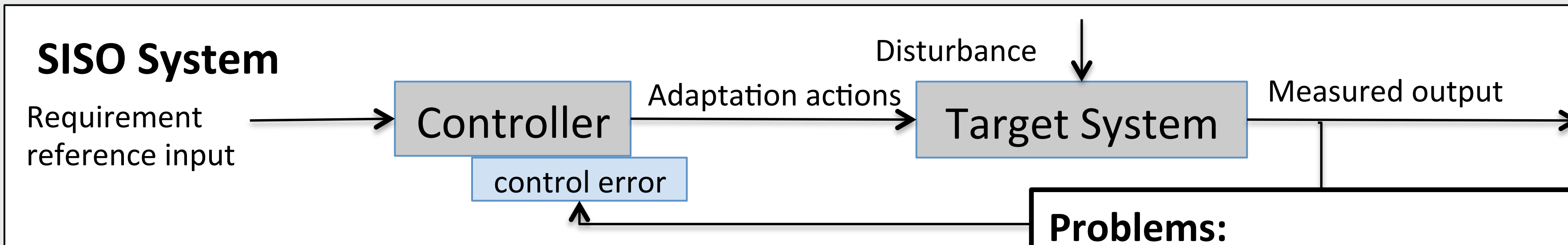


Mitigate the conflict with additional refinement

OK, I got all the variables
I need (maybe) and now?

Controlling Requirements

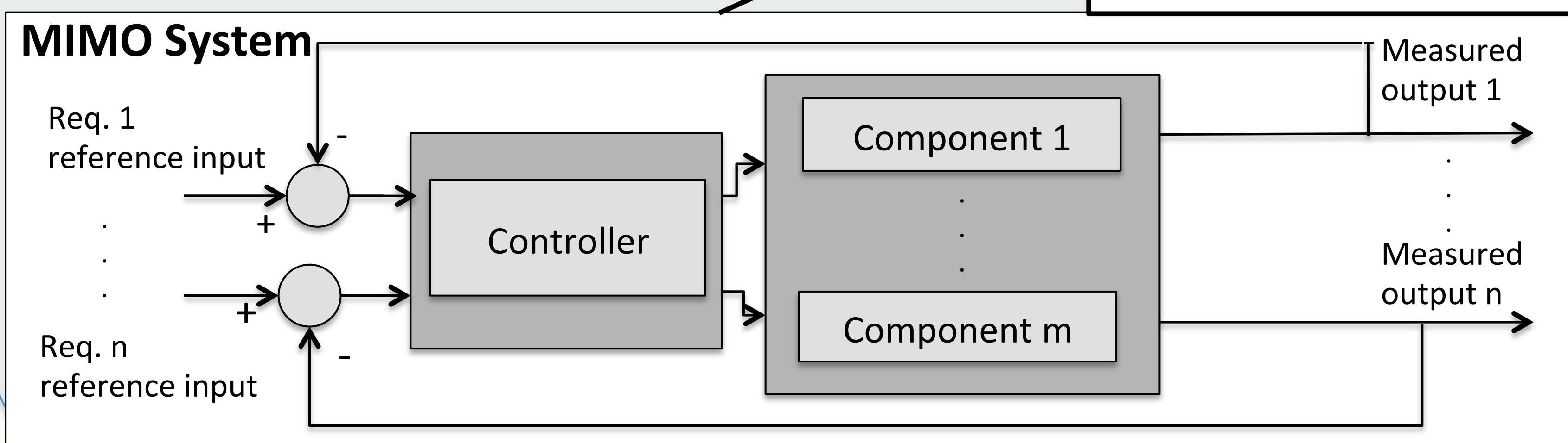
Initially [Cheng'09]:



Problems:

1. Conflicting requirements
2. Complex dynamics
3. Environmental disturbances

Reality [Angelopoulos'14]:

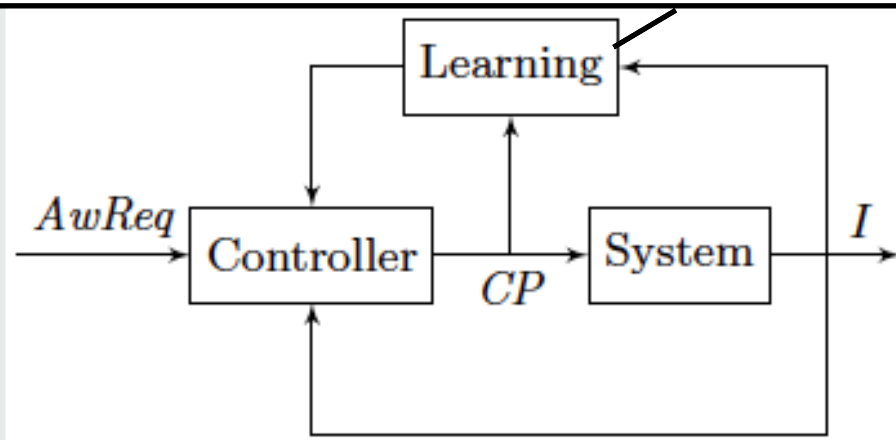


Model Predictive Control

Learn how environmental parameters behave over time and develop better adaptation plans

System Dynamics: $x(t+1) = A \cdot x(t) + B \cdot CP(t)$

$I(t) = C \cdot x(t)$



$$\text{minimize}_{CP_{t+k}} \quad \sum_{k=0}^{N-1} J(AwReq_{t+k}, I_{t+k}, CP_{t+k})$$

$$\begin{aligned}\text{subject to} \quad & I_{\min} \leq I_{t+k} \leq I_{\max} \\ & CP_{\min} \leq CP_{t+k} \leq CP_{\max} \\ & x_{t+k+1} = A \cdot x_{t+k} + B \cdot CP_{t+k} \\ & I_{t+k} = C \cdot x_{t+k} \\ & x_t = x(t), \quad k = 0, \ldots, N-1.\end{aligned}$$

The model that describes the system's behavior is derived either from a reliable simulation or the system itself.

Minimize the control error for every indicator according to its priority and the effort required (e.g. adding more meeting rooms requires more effort than increasing the allowed conflicts)

[Angelopoulos15b]

Contributions and Future Work

- A Three-Peaks modeling approach that captures every aspect of variability in software systems and their environments
- A control-theoretic approach that provides formal guarantees about its precision
- Stakeholders have an active role in the adaptation process
- More case studies from various domains to evaluate the generality of MPC and the effectiveness of the Three-Peaks approach (ongoing work)
- Develop tools to facilitate the analytical model derivation and the controller design (ongoing work)

References

[Nuseibeh'01]: B.Nuseibeh. Weaving Together Requirements and Architectures.

[Souza'11a] V.E.S. Souza, J. Mylopoulos. From awareness requirements to adaptive systems: A control-theoretic approach.

[Souza'11b] V.E.S. Souza, A. Lapouchnian, J. Mylopoulos. System Identification for Adaptive Software Systems: A Requirements Engineering Perspective.

[Ang'14] Dealing with multiple failures in Zanshin: A control-theoretic approach.

[Ang'15a] K. Angelopoulos, V.E.S. Souza, J. Mylopoulos. Capturing Variability in Adaptation Spaces: A Three-Peaks Approach.

[Ang'15b] K. Angelopoulos, A. Papadopoulos, J. Mylopoulos. Adaptive Predictive Control for Software Systems

References

[Cheng'09] B.H.C. Cheng et al. Software Engineering for Self-Adaptive Systems.

Thank You!

Questions?