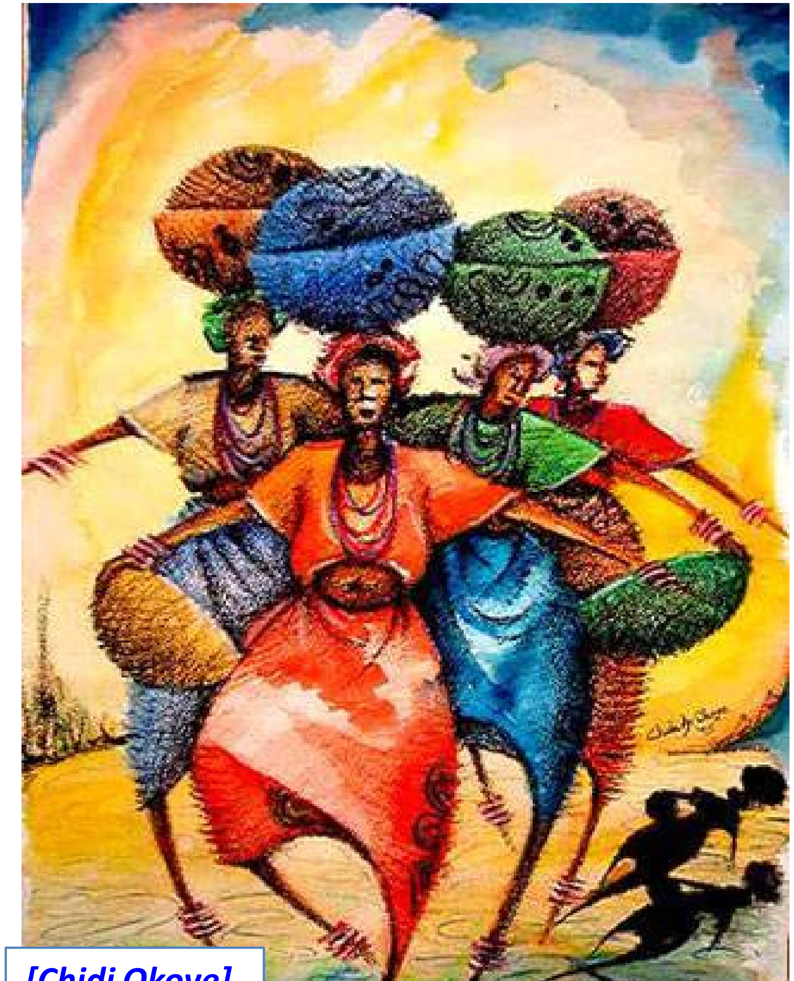
The Requirements Problem Revisited

John Mylopoulos
University of Trento

RE seminar, DISI
Trento, April 5, 2012



[Chidi Okoye]

Abstract

Jackson&Zave defined in [Jackson95] the requirements problem and gave an example of how it can be solved. At approximately the same time, van Lamsweerde [Dardenne93] and others proposed a goal-oriented view of requirements that characterized better the nature and the size of the problem. Much of the research that we have done in the past 15 years can be viewed as an extension of what was proposed in these two seminal papers intended to account for preferences, adaptivity, evolution, and more. We review some of this work, justify our proposed extensions, and suggest directions for further research.

Acknowledgements

- This presentation is based on on-going joint work with Alex Borgida (Rutgers), Neil Ernst (Toronto/UBC) and Ivan Jureta (Namur) on the requirements modeling language Techne. The presentation also touches on our recent work on adaptive software systems, with Alexei Lapouchnian and Vitor Souza (Trento).

The Requirements Problem (J&Z)

- In its original formulation [Jackson95], a requirements problem consists of finding a ***specification*** S for a given set of ***requirements*** R and ***indicative environment properties*** E such that

$$E, S \mid\!- R$$

meaning: "… satisfaction of the requirements can be deduced from satisfaction of the specification, together with the environment properties…" [Jackson95]

- We prefer a formulation where environment properties are replaced by ***domain assumptions*** (D) and inference is replaced by entailment

$$D, S \models R$$

Problem refinement

- (Akin to program refinement) Start with requirements and keep refining them to eliminate mention of non-executable elements.
- For instance, (with slot ≡ no-conflict timeslot)

"Schedule a meeting" (Req1)

"Find a timeslot t when all participants are free" and
"find free room r for timeslot t" and "book(r,t)" (Spec1)

"For any slot there are free mtg rooms" (DA1)

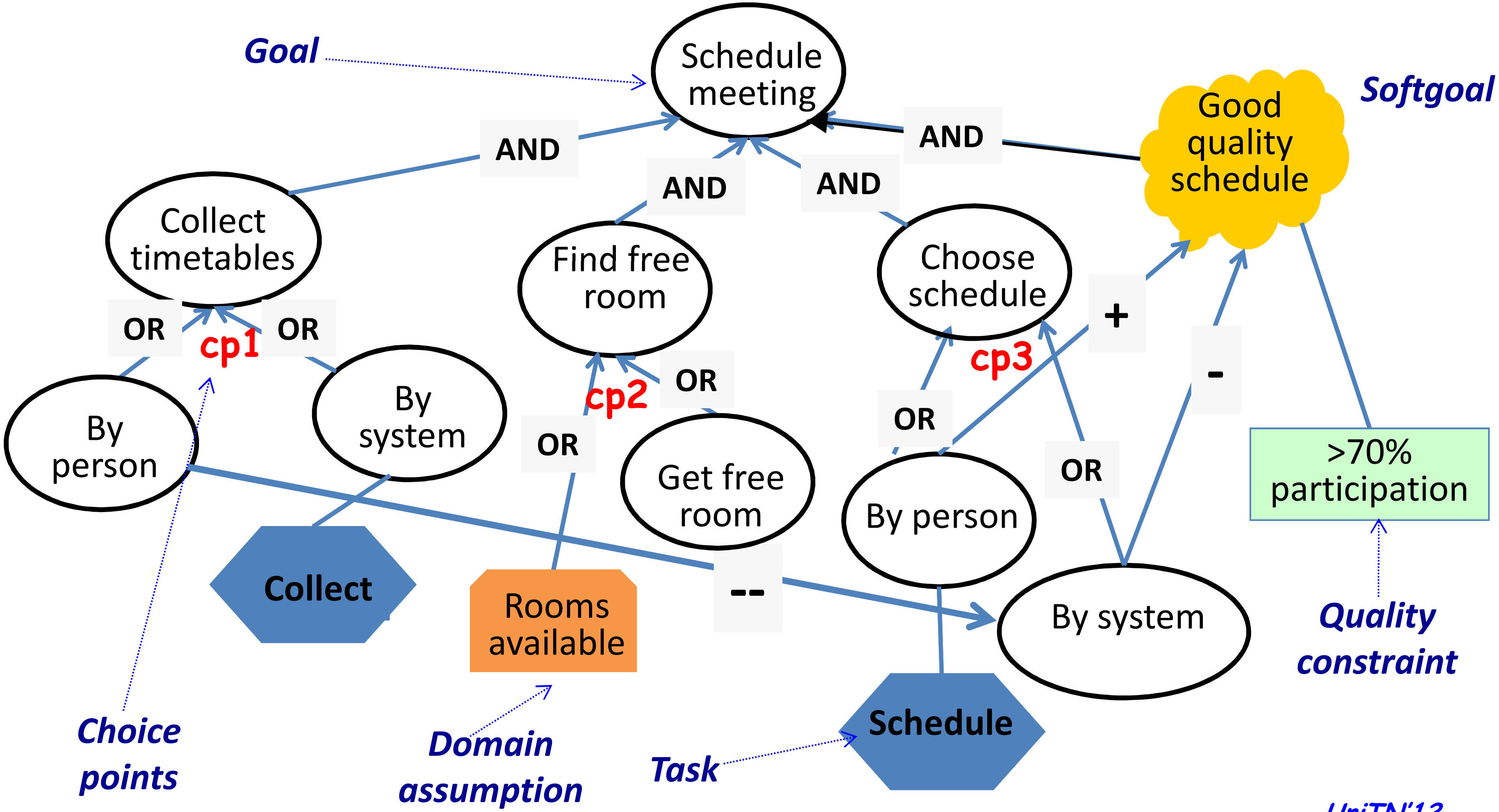
"There are always slots for any mtg request" (DA2)

Spec1 э DA1 э DA2 ® Req1 (DA0)

Spec1 э DA0 э DA1 э DA2 |= Req1

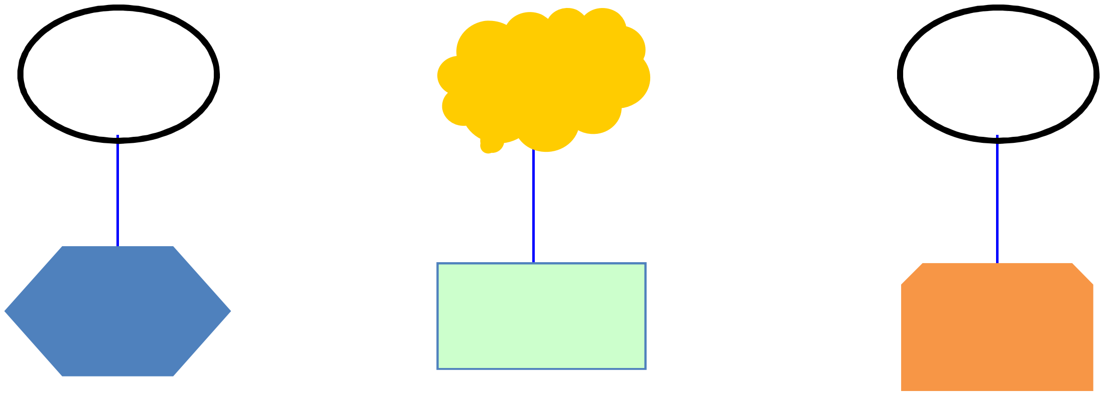
Requirements as Goals (GORE)

- Requirements are now goals and (requirements) problem solving amounts to incremental goal refinement.



Operationalizations

- We are using three types of operationalizations here.
- The first two are proactive, the last ***opportunistic***.

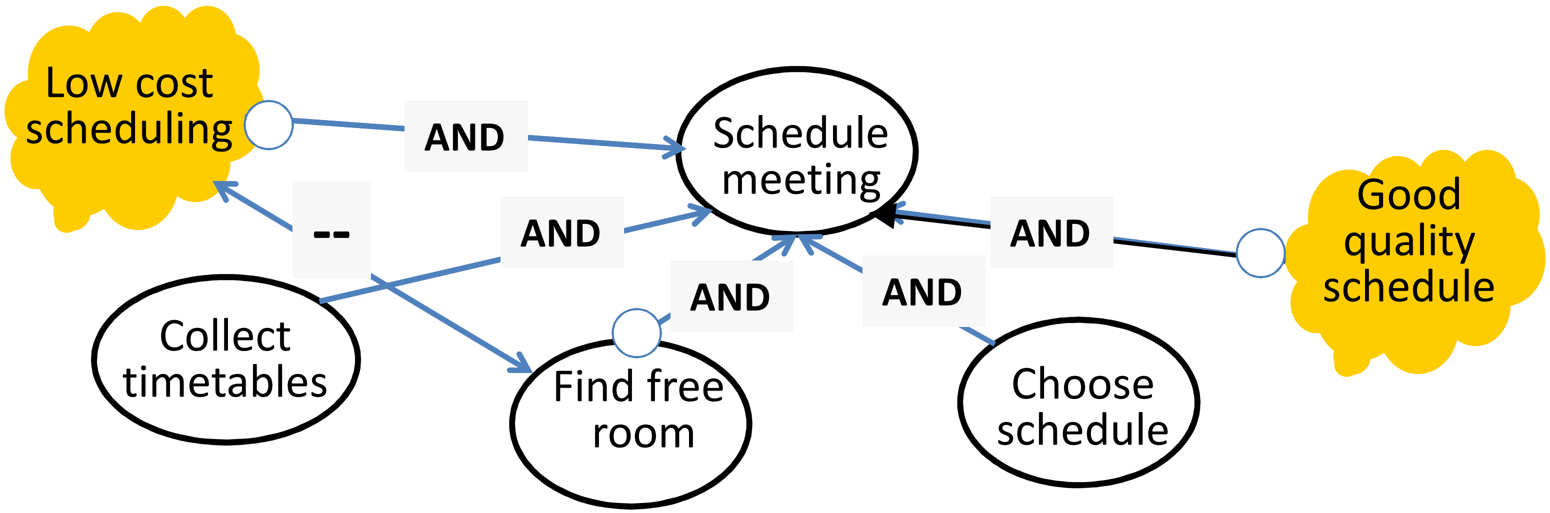


Requirements as Goals

- Here, specifications consist of tasks, domain assumptions and quality constraints that together satisfy requirements, e.g., for G:ScheduleMtg, one specification is {T:Collect, T;Schedule, D:RoomsAv, QC: ‘>70% participation’}
- Unlike J&Z, goal refinement generates ***a space of possible specifications*** and the requirements problem amounts to finding those that satisfy R.
- The GORE version of the requirements problem can be reduced to SAT solving [Sebastiani04].
- This algorithm constitutes the backbone of goal-oriented requirements problem solving.

Preferences & Priorities (P&P)

- Preferences are "nice-to-have" requirements. Among them, there can be binary priority relations.



- Low cost >> Find free room (priorities)
- Low cost >> Good quality schedule

Preferences & Priorities

- Now a solution consists of a specification that satisfies all mandatory goals and a maximal consistent subset of preferred ones, with no better solution wrt priorities.
- The requirements problem is now an optimization problem, rather than merely a satisfaction one.
- Note: The semantics of preferred requirements are different from those of optional features.

  e.g., PR1 = prefs {R1,R2} has ?? solutions

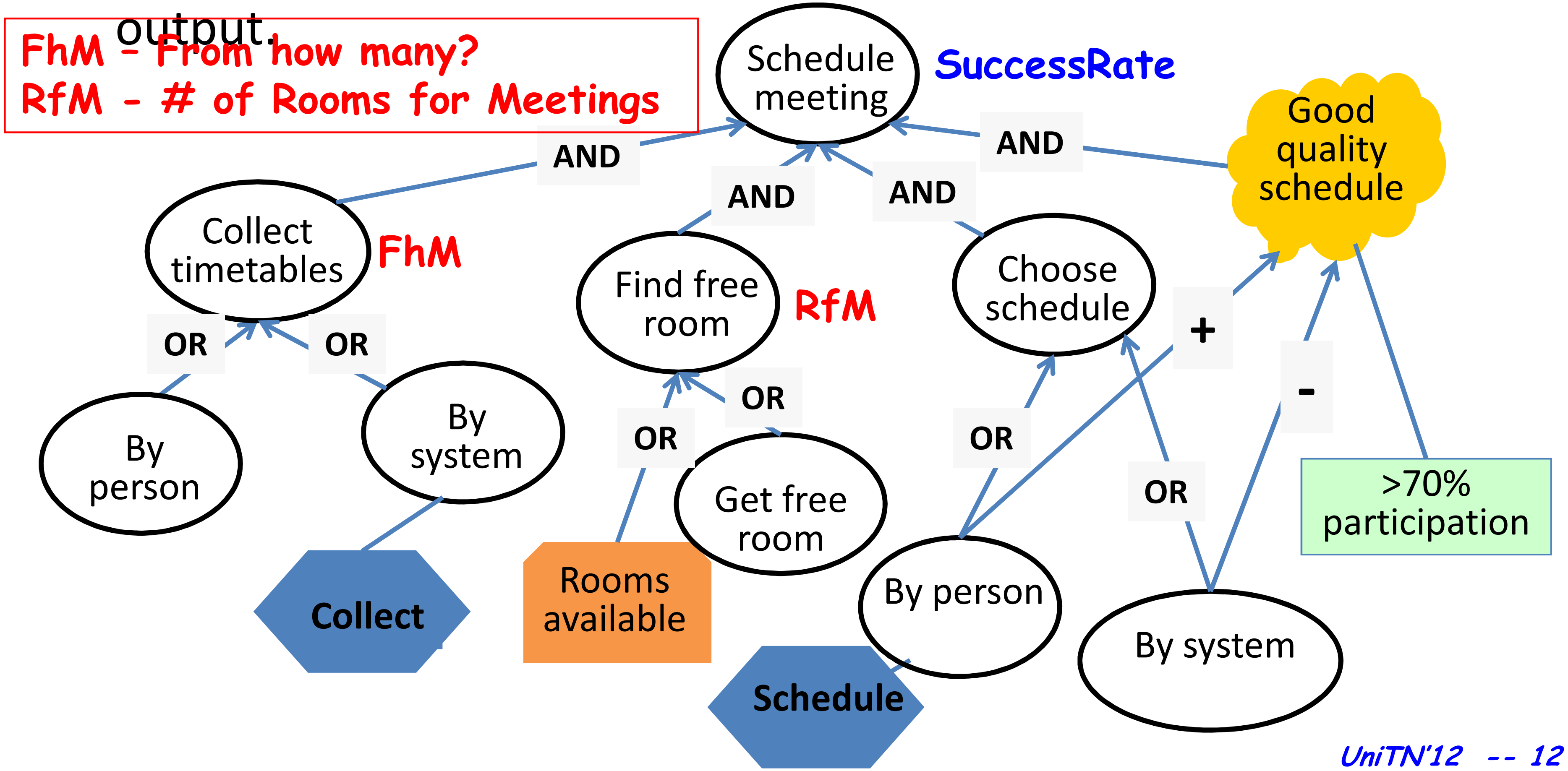
  PR2 = optional {F1,F2} has ?? solutions

Preferences & Priorities

- One way to tackle this version of the requirements problem is to adopt ideas from AI planning. However, AI planners are very expressive, and many of their features are best used during design, rather than RE [Liaskos10].
- Another way is to develop algorithms from first principles [Jureta10].
- In either case, intractability here is a given, while scalability is an open question (but see [Ernst10]).

Control Variables and Indicators (CV&I)

- Success/failure of a specification to fulfill its requirements depends on ***control variables*** and ***indicators*** (gauge variables)that determine resp. resource allocation, quality of output.



Control Variables and Indicators

- To define the requirements problem here, we also need to define how indicators depend on control variables,

  e.g., F(FhM,RfM,SuccessRate,CostPerMtg) = 0

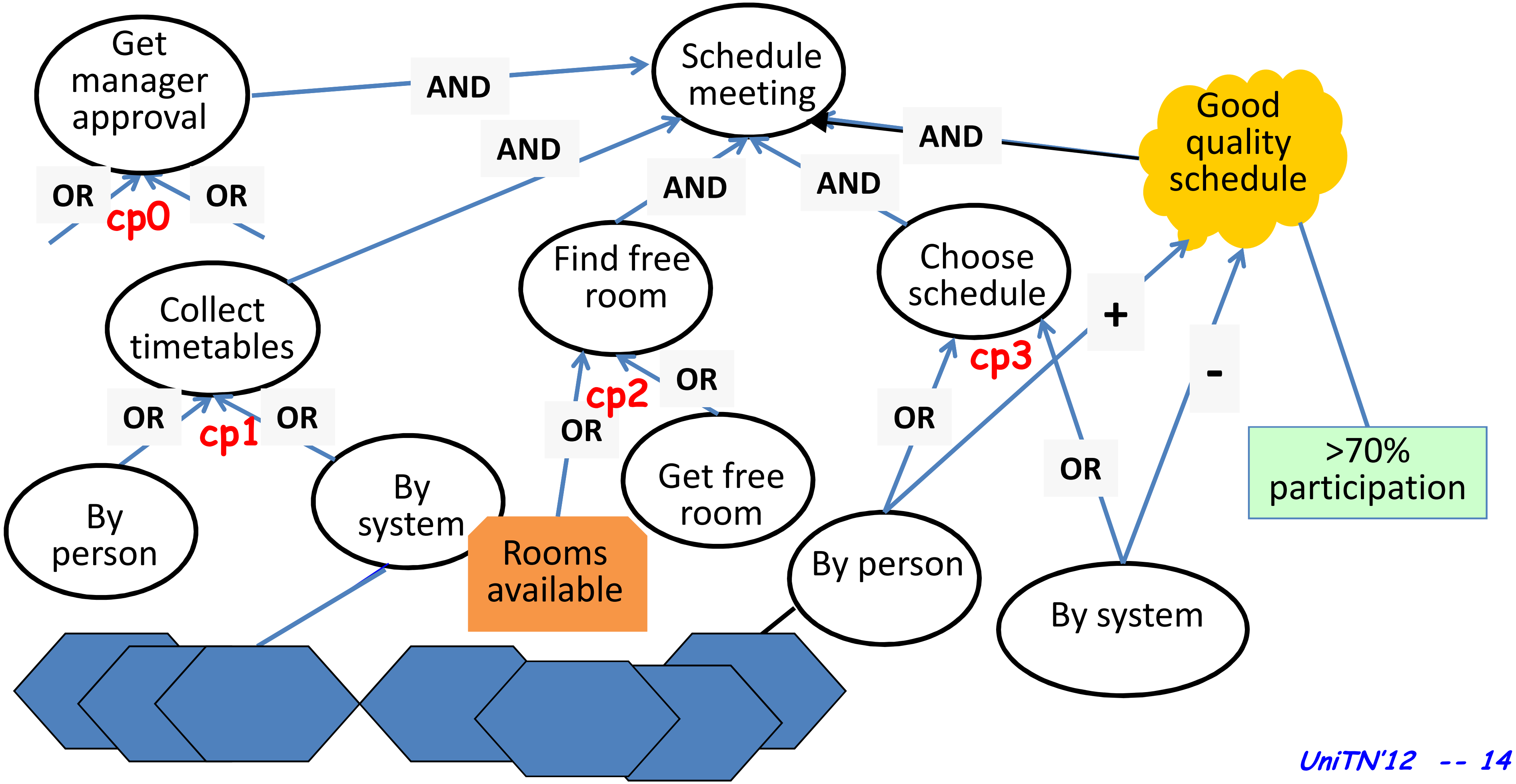
  G(FhM,RfM,SuccessRate,CostPerMtg) ≥ 0

  min[H(SuccessRate,CostPerMtg)]

- Now we have an optimization problem that seems to lie somewhere between OR-style optimization and SAT solving …

The Incremental Requirements Problem

- Suppose now we have an architecture that implements several specifications and a running (=old) solution, and a requirement changes …



The Incremental Requirements Problem

- All we need to do is run our GORE/P&P/CV&I/… algorithm for solving the requirements problem, right? …
- Not quite, if we want to:
  - Maximize familiarity – use as much as possible of the old solution (user perspective)
  - Minimize effort – minimize the number of tasks that need to be implemented (vendor perspective)
- We seem to need algorithms here that use a minimum repair principle to find new solutions.
- [Ernst11] studies a class of such algorithms using AI Truth Maintenance Systems (ATMS).

Qualitative Adaptive Control

- For the CV&I version of the requirements problem, it is clearly impractical to assume that you can have functions F, G that relate control variables and indicators.
- So, let's assume instead that control variables and indicators are related through qualitative differential constraints, such as

$$\Delta \frac{\text{SuccessRate}}{\text{RfM}} (1,10) > 0 \qquad \Delta \frac{\text{SuccessRate}}{\text{RfM}} (10,*) = 0$$

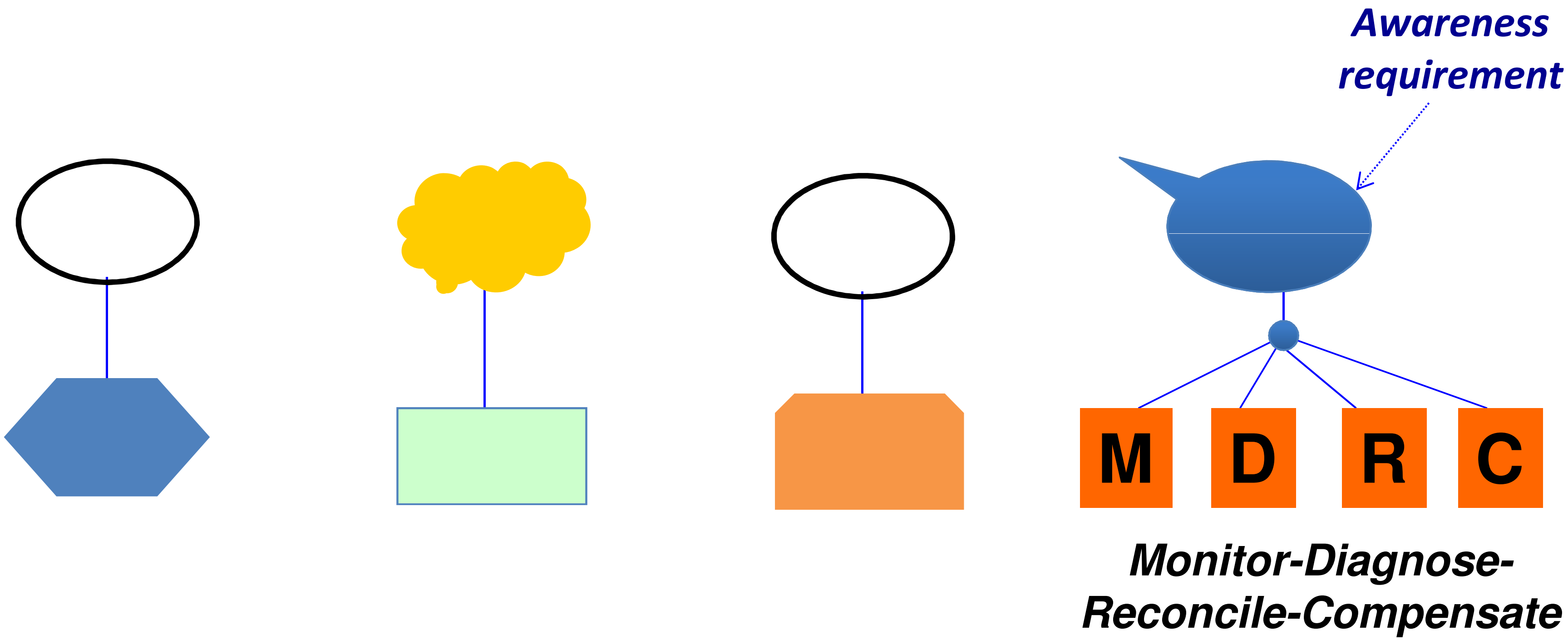
$$\Delta \frac{\text{SuccessRate}}{\text{FhM}} (60,100) < 0 \qquad \Delta \frac{\text{SuccessRate}}{\text{FhM}} (0,60) = 0$$

Qualitative Adaptive Control

- Similar qualitative constraints were used in AI since the 80s to achieve qualitative simulation of physical systems (e.g., flushing a toilet).
- We want to use such constraints for system identification of adaptive software systems [Souza11]: characterize the controllability space for a software system, defined in terms of a requirements model, choice points, control variables and indicators with qualitative differential constraints linking all of the above. …

Operationalizations revisited

- We now use four types of operationalizations, two proactive, one opportunistic, the last ***reactive***.



Summary

- The requirements problem manifests itself in many forms and variations thereof.
- Finding solutions to the problem in its many manifestations calls for a fusion of SAT- and SMT-solving and optimization techniques that goes beyond the state-of-the-art and needs study.
- Such solutions constitute a useful baseline for research on adaptive software systems.
- More importantly perhaps, such solutions can serve as a starting point for a Theory of RE.

References

- [Dardenne93] Dardenne, A., van Lamsweerde, A. and Fickas, S.,"Goal-Directed Requirements Acquisition", in The Science of Computer Programming 20, 1993.
- [Ernst10] Ernst N., Borgida A., Jureta I., Mylopoulos J., "Reasoning with Optional and Preferred Requirements", 29th International Conference on Conceptual Modeling (ER'10), Vancouver, Nov. 2010.
- [Ernst11] Ernst N., Borgida A., Mylopoulos J., Jureta I., "Finding Incremental Solutions for Evolving Requirements" (submitted for publication).
- [Feather98] Feather, M., Fickas, S., van Lamsweerde, A., Pounsard "Reconciling System Requirements and Runtime Behavior", 9th International Workshop on Software Specification and Design (IWSSD'98), Isobe Japan, April 1998.
- [Finkelstein09] Finkelstein, A., Harman, M., Mansouri, S., Ren, J., Zhang, Y.: A search based approach to fairness analysis in requirement assignments to aid negotiation, mediation and decision making. Requirements Engineering J. 14(4), 231-245.
- [Jackson95] Jackson M., Zave P., "Deriving Specifications from Requirements: An Example", 17th International Conference on Software Engineering (ICSE'95).
- [Jureta10] Jureta, I., Borgida, A., Ernst, N., Mylopoulos, J., "Techne: Towards a New Generation of Requirements Modeling Languages with Goals, Preferences and Inconsistency Handling", 19th Int. IEEE Conference on Requirements Engineering (RE'10), Sydney Australia, Sept. 2010.

References

- [Liaskos10] Liaskos, S., McIlraith, S., Sohrabi, S., Mylopoulos, J., “Integrating Preferences into Goal Models for Requirements Engineering”, 19th Int. IEEE Conference on Requirements Engineering (RE’10), Sydney Australia, September 2010.
- [Sebastiani04] Sebastiani R., Giorgini P., Mylopoulos J., “Simple and Minimum-Cost Satisfiability for Goal Models”, 16th International Conference on Advanced Information Systems Engineering (CAiSE’04), Riga, June 2004, Springer-Verlag LNCS 2003, 20-35.
- [Souza11] Souza V., Lapouchnian A., Mylopoulos J., “The System Identification Problem for Adaptive Software Systems”, (unpublished manuscript).
- [vanLamsweerde09] van Lamsweerde A., “Reasoning About Alternative Requirements Options”, in Borgida A., Chaudhri V., Giorgini P., Yu E., Foundations and Applications of Conceptual Modelling, Springer-Verlag LNCS 5600, June 2009.
- [Chidi Okoye] http://www.modernartimages.com/expressionsofdance3.htm