# The Requirements Problem in Software Engineering

John Mylopoulos
University of Trento

Symposium on Applied Computing (SAC)
Pisa, April 5, 2016

# Acknowledgements

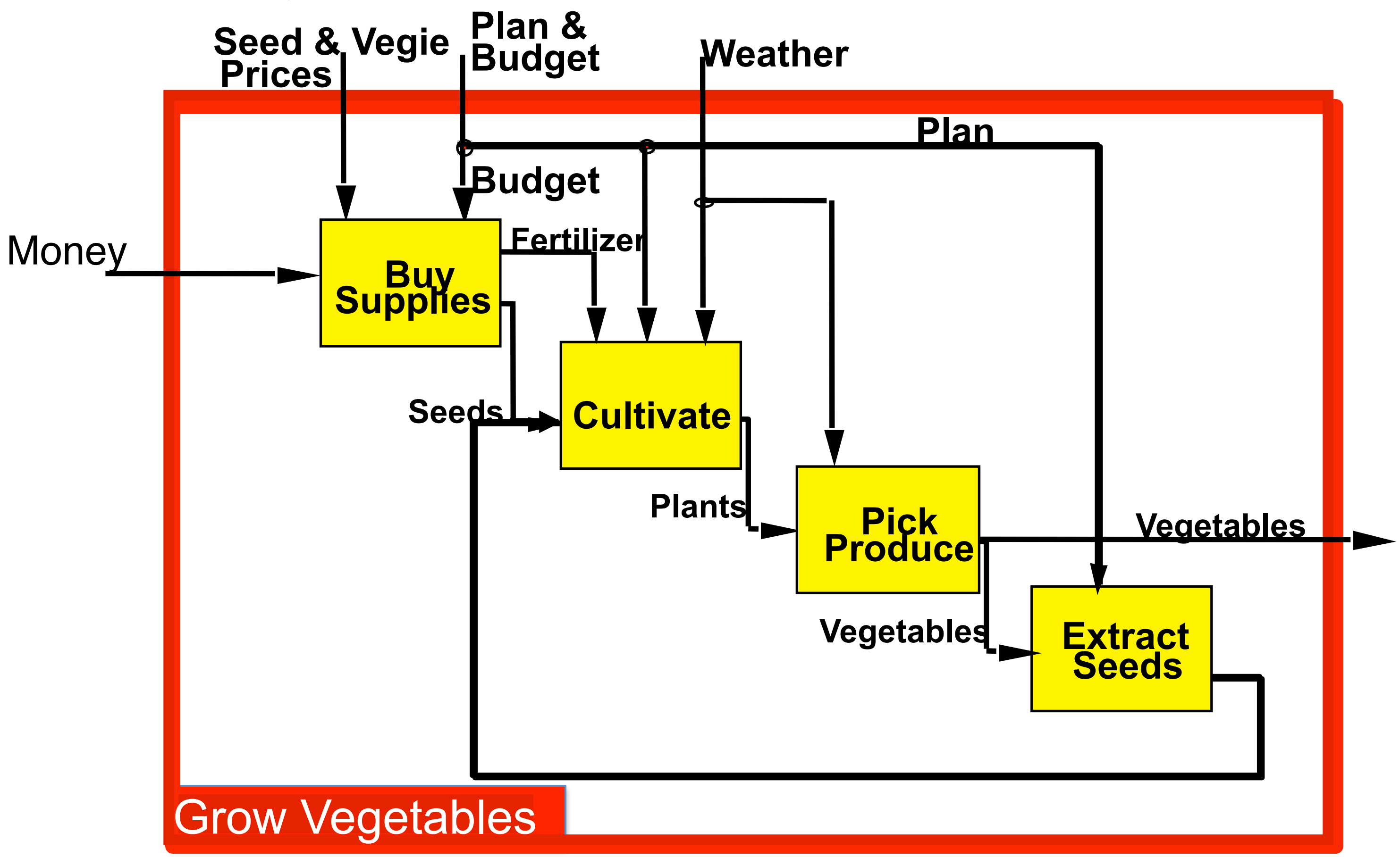
- The presentation touches on research conducted over the past decade at the Universities of Trento and Toronto with colleagues and students, spanning multiple PhD theses, including Neil Ernst (Toronto), Alex Borgida (Rutgers, USA), Ivan Jureta (Namur, Belgium), Vitor Souza (Espirito Santo, Brazil), Roberto Sebastiani, Paolo Giorgini, Kostas Angelopoulos, Fatma Aydemir, Mai Chi Nguyen (Trento, Italy).

# Requirements Engineering (RE)

- Concerned with the elicitation, analysis and refinement of stakeholder requirements in order to produce a specification for a system-to-be.
- Founded on seminal works by Douglas Ross, Michael Jackson and others in the mid-70s.
- Unique research area within CS because its task is not to solve problems, but rather to ***define*** ones.
- Interesting area, because stakeholder ("early") requirements are necessarily vague, informal, self-contradictory, and more (... in short, "scruffy"), but they are requirements none-the-less.

# Origins

- Requirements are activities/functions the system-to-be will perform within its operational environment (Douglas Ross, c.1977).



# (Requirements) Specifications

- Back then, the requirements problem consisted of generating a list of functional requirements -- a ***specification*** -- for the system-to-be.
- Much research in the 70s and 80s focused on making such specifications formal.
- An IEEE standard was established in 1993, [IEEE93] where requirements specifications were prescribed to include functions and metricized quality constraints, such as "Users shall be able to use the system after 3hrs of training".
- … And that was the state-of-affairs in the mid-90s …

# RE is both hard and expensive

- Major cause of project failure: Survey of US software projects by the Standish group

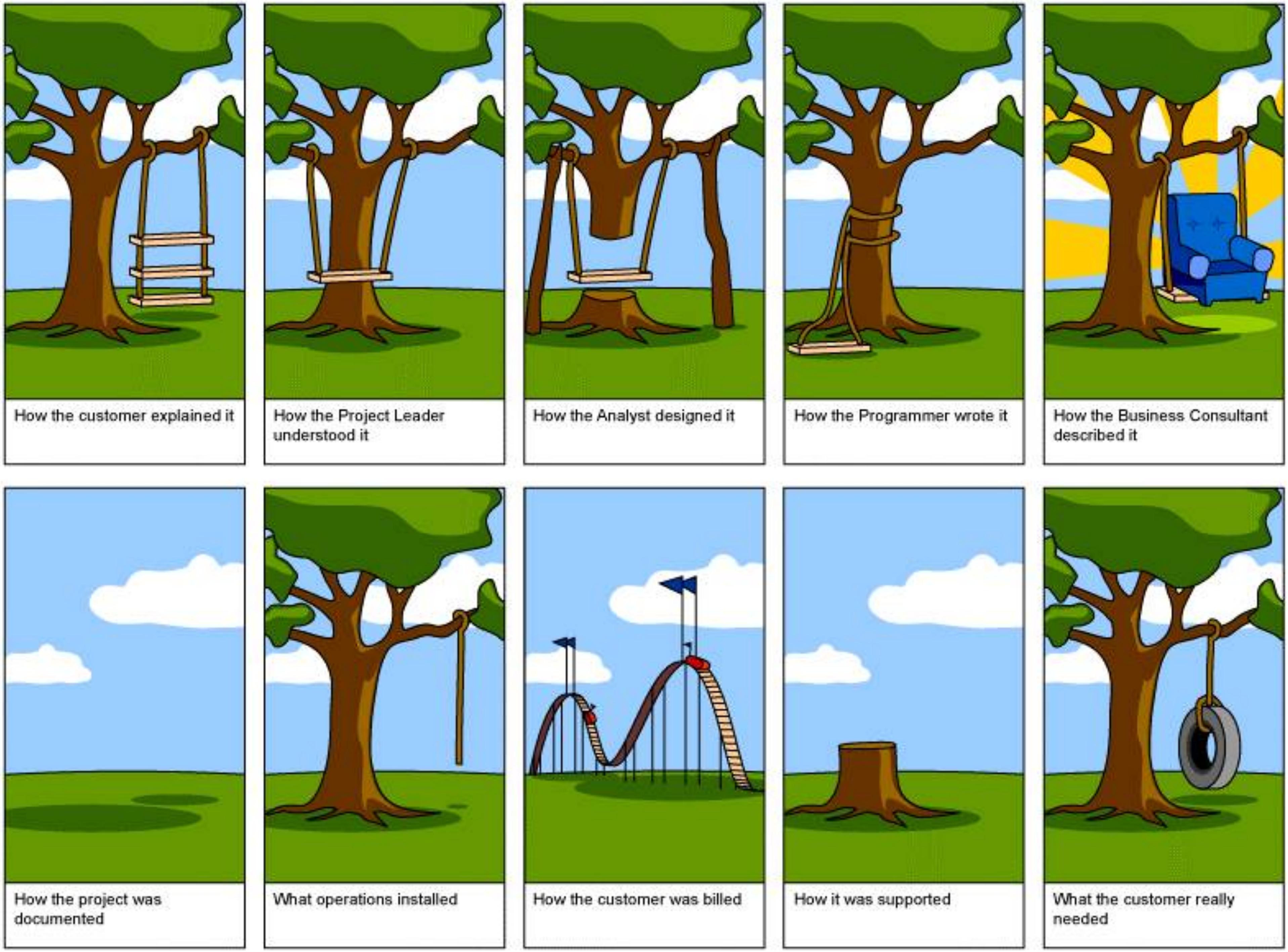
| | 1994 | 1998 |
|---|---|---|
| Successful | 16% | 26% |
| Challenged | 53% | 46% |
| Cancelled | 31% | 28% |

**Top 3 success factors:**
1) User involvement
2) Executive management support
3) Clear statement of requirements

**Top 3 factors leading to failure:**
1) Lack of user input
2) Incomplete requirements & specs
3) Changing requirements & specs

(see also http://calleam.com/WTPF/?page_id=1445)

- Cost of fixing errors: A requirements error found during testing costs 100 times more than a programming error found during testing.


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

# The requirements problem

- The requirements elicited from the stakeholders are statements of needs.
- The ***requirements problem*** is to transform these needs through a systematic process into a specification consisting of functions, quality constraints, and domain assumptions.
- I'll go over a series of formulations of this problem and discuss the scalability of solutions that have been proposed in the literature.

# 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 ≡ timeslot)

“Schedule a meeting” (Req1)

findSlot(t,PartL), findRoom(r,t), book(r,t) (Spec1)

“For any slot there are free mtg rooms” (DA1)

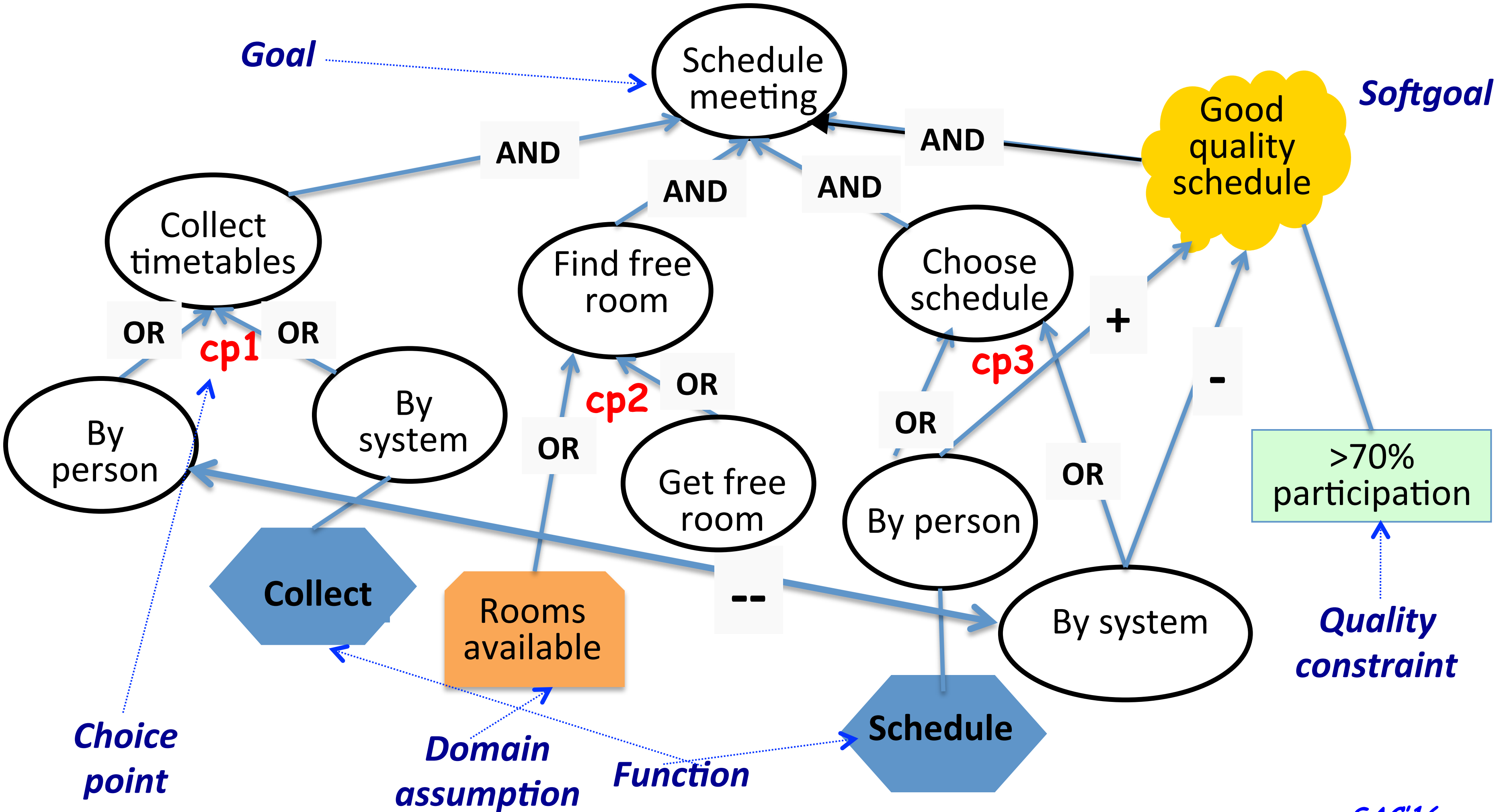
“For any list of participants there is a slot when they are free” (DA2)

$$\text{Spec1} \wedge \text{DA1} \wedge \text{DA2} \Rightarrow \text{Req1} \qquad \text{(DA0)}$$

$$\text{Spec1}, \{\text{DA0},\text{DA1},\text{DA2}\} \models \text{Req1}$$

# Requirements as goals (GORE)

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

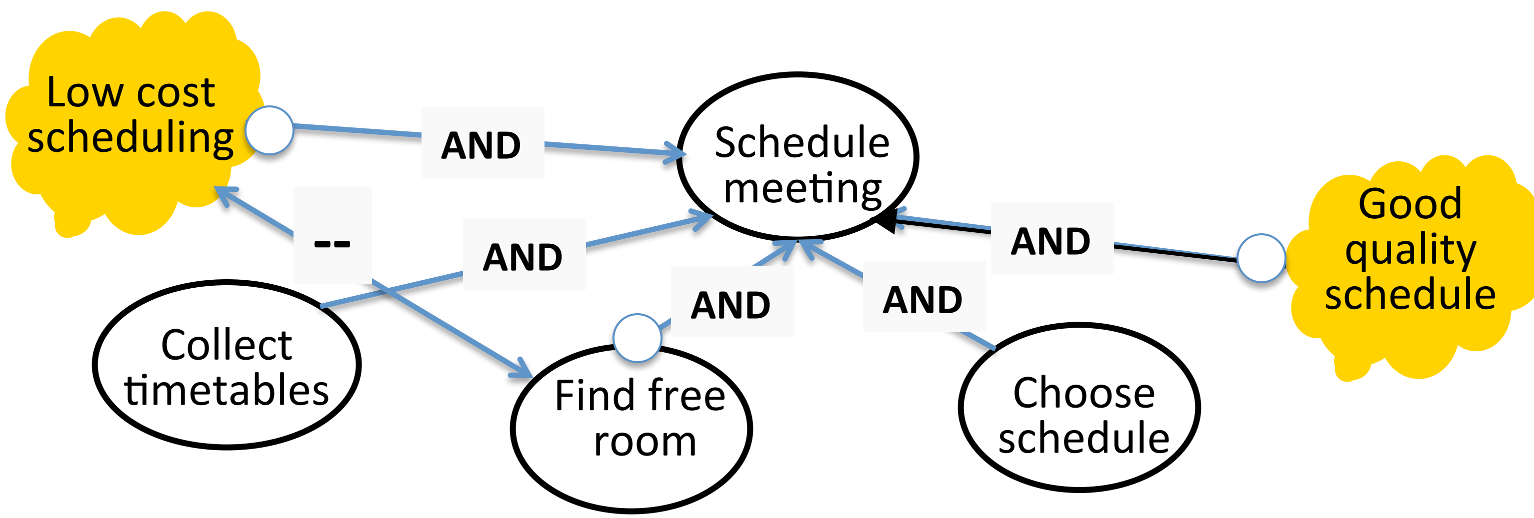


# Requirements as goals

- Here, specifications consist of functions, domain assumptions and quality constraints that together satisfy requirements, e.g., for G:ScheduleMtg, one specification is {F:Collect, F;Schedule, DA: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] and scales for goal models of size ***O***(1K).

# Preferences and 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 and priorities

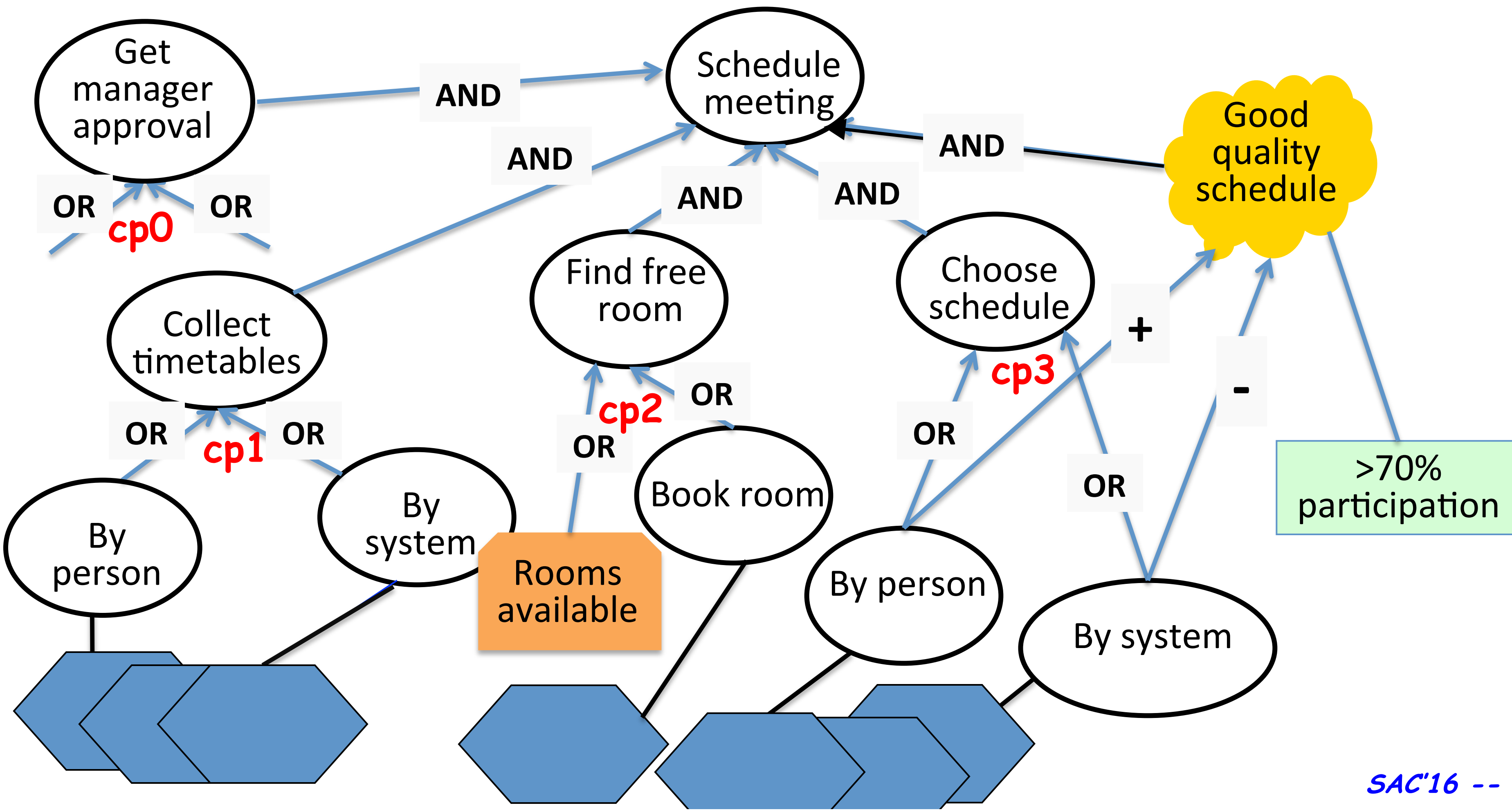
- Now a solution consists of a specification that satisfies all mandatory goals and a maximal subset of preferred ones, with no better solution wrt priorities.
- The requirements problem is now an optimization problem, rather than merely a satisfaction one.
- ***Note***: The semantics of preferred requirements are different from those of optional features. For example, if I have a problem consisting of two consistent requirements {R1,R2}, there is one solution if R1, R2 are preferred, but 4 solutions if they are optional.

# Preferences and priorities

- One way to tackle this version of the requirements problem is to adopt ideas from AI planning. However, several features of AI planners 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.
- [Ernst10] uses local search techniques with good performance results to improve on naïve search.

# The incremental requirements problem

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

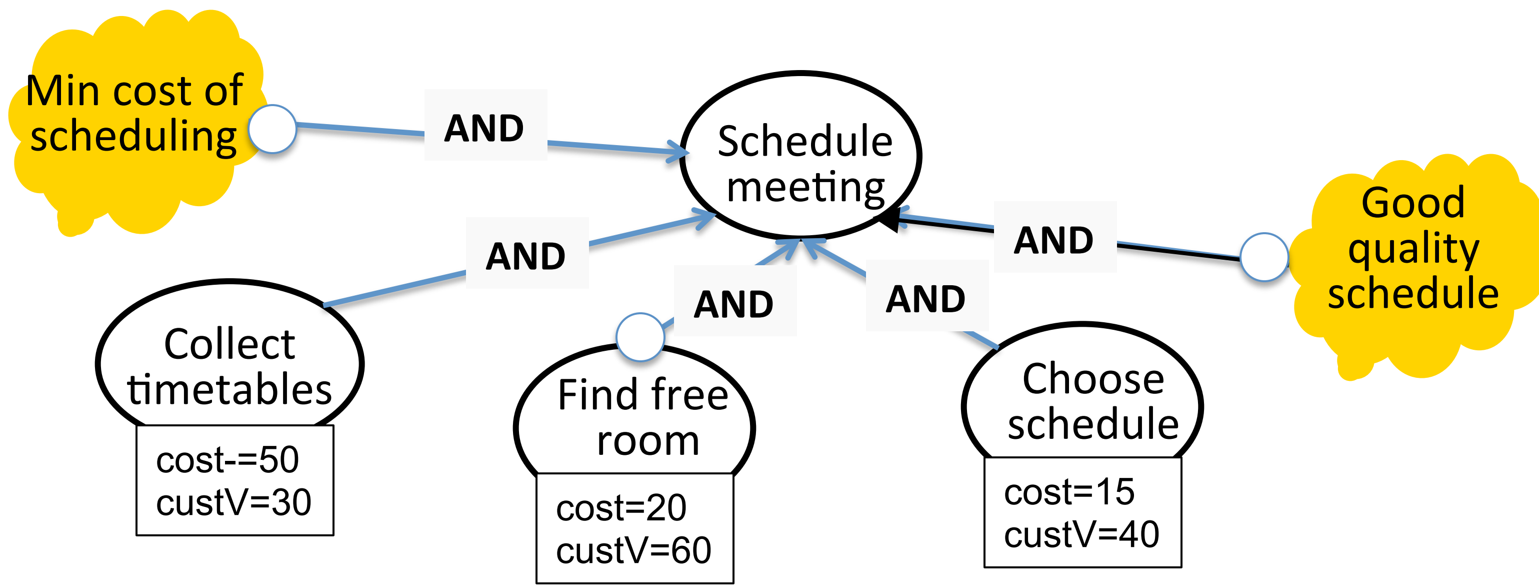


# The incremental requirements problem

- All we need to do is run our GORE/P&P/… search algorithm for solving the ***new*** requirements problem, right? …
- Not quite, if we want to:
  - ✓ Maximize familiarity – use as much as possible the old solution (user perspective)
  - ✓ Minimize effort – minimize the number of functions that need to be implemented (vendor perspective)
- We need algorithms here that search for repairs when the requirements problem "breaks" due to new requirements.
- [Ernst11] studies a class of such algorithms using AI Truth Maintenance Systems (ATMS).

# Requirements problem cum optimization

- Suppose now we further extend our requirements language to associate costs, customer value etc. and other attributes to each goal, and also allow optimization (min/max) goals



# Requirements problem cum optimization

- We need now search engines with richer native languages than SAT solvers.
- Chi Mai Nguyen [Nguyen15] is experimenting with Optimization Modulo Theories/Satisfiability Modulo Theories (aka OMT/SMT) solvers for solving such problems.
- In particular, she is using OptiMathSAT [Sebastiani15] which supports linear arithmetic over rationals, as well as optimization over multiple objective functions, either singularly or lexicographically.
- Our expectation is that her solution will be scalable.

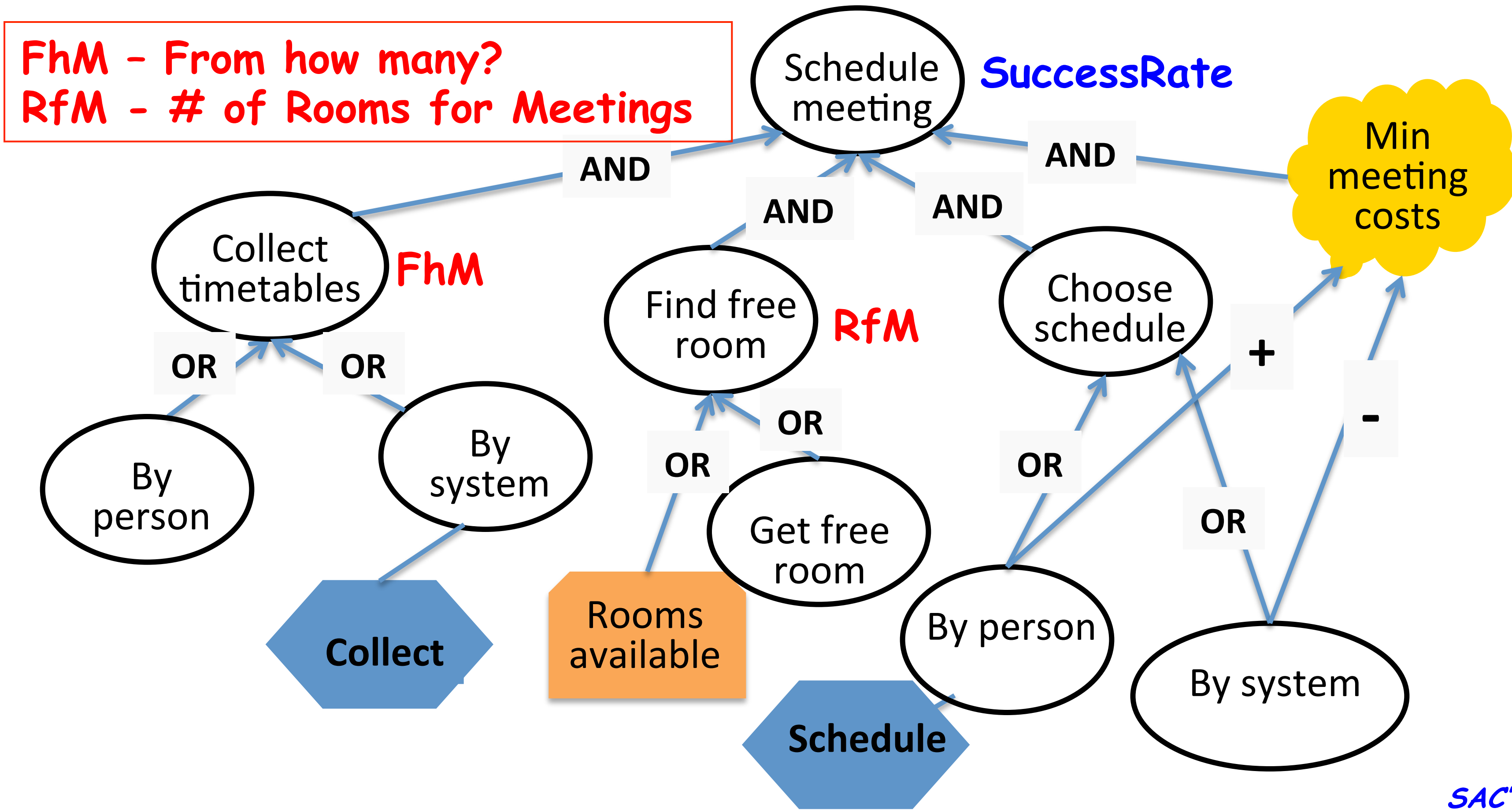
# The next release problem

- This is a variant of the incremental requirements problem: Given an existing system and a set of new mandatory and/or preferred requirements, you want to find a specification for the next release of the system that optimizes several objective functions.

- Fatma Başak Aydemir [Aydemir15] is working on this problem, using the OptiMathSAT solver as search backend.

# The adaptation problem

A specification fulfill its requirements depending on ***control variables*** and ***indicators*** that determine respectively resource allocation, quality of output, etc.



# The adaptation problem

- To define the adaptation problem, 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

- Now suppose that the system is monitoring its performance and some requirements are failing (or, equivalently, some indicators are out of bounds).
- We need to find a new set of values for control variables that "restores" failed requirements in an optimal way.
- Again, this seems like a search problem that lies somewhere between OR-style optimization and SAT solving ...
- Kostas Angelopoulos is experimenting with an OptiMathSAT-based tool to solve such problems.

# Summary

- The requirements problem manifests itself in many forms and variations thereof.
- Finding scalable solutions to the problem in its many manifestations calls for a fusion of SAT/SMT solving and optimization techniques.
- Such solutions constitute a useful baseline for research on RE, but also Adaptive Software Systems.
- More importantly perhaps, such solutions could serve as starting point for a Theory of RE.

# References


- [Aydemir15] Aydemir F., Giorgini P., Mylopoulos J., "Solving the Next Release Problem through Qualitative Reasoning", to be submitted for publication.
- [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., 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.
- [Harman01] Harman M., Jones B., "Search-Based Software Engineering", *Information and Software Technology 43*, 2001.
- [IEEE93] IEEE Computer Society. Software Engineering Standards Committee and IEEE-SA Standards Board, IEEE recommended practice for Software Requirements Specifications, Institute of Electrical and Electronics Engineers, 1993.
- [Jackson95] Jackson M., Zave P., "Deriving Specifications from Requirements: An Example", 17th International Conference on Software Engineering (ICSE'95).

# References


- [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.
- [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.
- [Nguyen15] Nguyen Chi Mai, Sebastiani R., Giorgini P., Mylopoulos J., "Multi-Objective Reasoning with Constrained Goal Models", submitted for publication.
- [Sebastiani04] Sebastiani R., Giorgini P., Mylopoulos J., "Simple and Minimum-Cost Satisfiability for Goal Models", 16th Int. Conference on Advanced Information Systems Engineering (CAiSE'04), Riga, June 2004, Springer-Verlag LNCS 2003, 20-35.
- [Sebastiani15] Sebastiani R., Tomasi S., "Optimization Modulo Theories with Linear Rational Costs", *ACM Transactions on Computational Logics 16(2)*, 2015 (to appear).
- [Souza11] Souza V., Lapouchnian A., Mylopoulos J., "The System Identification Problem for Adaptive Software Systems", 30th International Conference on Conceptual Modelling (ER'11), Brussels, Nov. 2011.
- [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.