

Evolution of Software and Research Infrastructures: A Report from the Trenches

John Mylopoulos
University of Trento

ECSS 2011, Milano ITALY
November 8, 2011



[Chidi Okoye]

Abstract

The first part of this presentation focuses on the main objectives and preliminary results of the Lucretius project (ERC advanced grant, 2011-16), whose aim is to develop concepts, techniques and tools for supporting software evolution. Evolution may involve adaptation of a system to a new behaviour that better copes with the system's operational environment, or mutation of a system to deal with unforeseen requirements (e.g., derived from a new regulation). Moreover, evolution may be automatic (e.g., self-adaptation), or manual, or something in-between.

The second half of this presentation focuses on academic research funding infrastructures in Europe, both indicative (as-they-are) and optative (as-they-ought-to-be), from the perspective of a researcher who recently moved from Canada to Europe (Italy).

Based partly on material from [Lucretius11].

I. Software evolution

- Known hard problem since the very beginning of Software Engineering (SE).
- Laws of software evolution put forward a theory of how software (code) evolves over its lifetime [Lehman80]
 - e.g., “The complexity of a program increases, unless ... “
- Software Reengineering, Software Maintenance study aspects of software evolution, and constitute important areas within SE -- largely code-focused.

Requirements-driven software evolution

- 🌈 We want to develop design principles for software that evolves gracefully, in response to changes to its requirements and/or its domain (= operational environment).
- 🌈 The principles need to address questions such as
 - What are the primitive concepts in terms of which we conceive and design such software?
 - What runtime support (with/without humans in the loop) do we need to enact/facilitate evolution?
- 🌈 Basic premise: Requirements models constitute (part of) the hereditary material (DNA) of software. Throughout its lifecycle, a software system consists of a requirements model and code that operationalizes the requirements (OK, and other things ...).

The Requirements problem (J&Z)

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

$$E, S \vdash R$$


meaning: “... satisfaction of the requirements can be deduced from satisfaction of the specification, together with the environment properties...” [Jackson95]

🌐 *Behaviours* are implementations of specifications.

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


$$D, S \models R$$

Evolution mechanisms

 **Adaptation**: In response to changes in the environment, system changes its behaviour – but not its requirements -- at runtime in order to continue to fulfill its requirements.


Change: $D \rightarrow D'$, Problem: $D', S'(S) \models R$

E.g., flight control system changes automatic->manual mode

 **Mutation**: System changes its specification in response to changes to its requirements.

Change: $R \rightarrow R'$, Problem: $D, S'(S) \models R'$

E.g., new requirements added because of a new law.

 **Composition (Reproduction)**: Requirements models of two systems are composed to produce a new requirements model within the same software “species”.

 ...

Requirements problem spaces

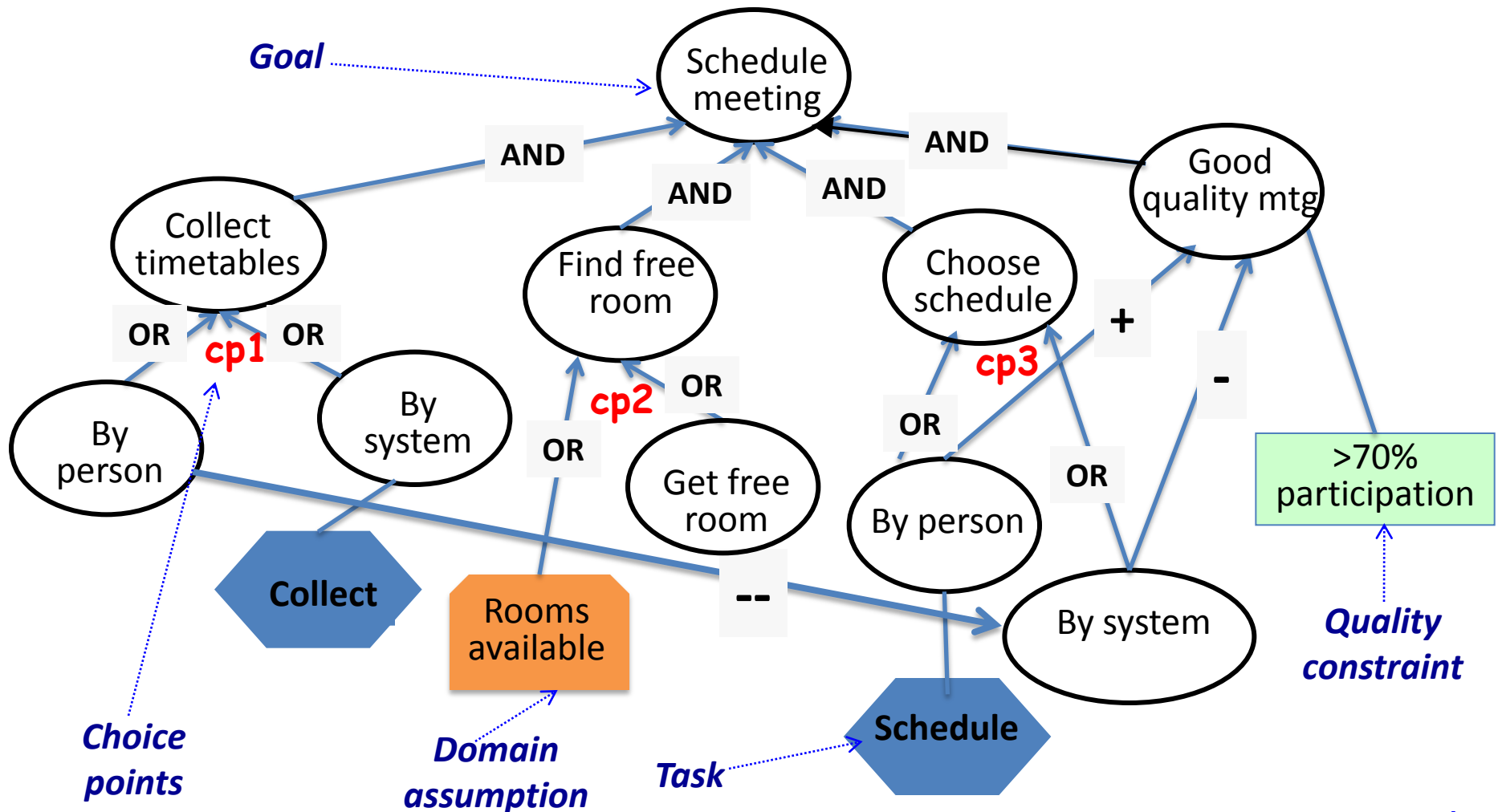
🌈 At the heart of every evolution there is a changed requirements problem.

🌈 Consequences:

- ✓ Need to characterize requirements problem spaces
- ✓ Need efficient algorithms for searching such spaces with different search criteria, e.g., (i) new solution should be best wrt new RP, (ii) new solution should be as similar to old as possible.

Requirements as goals (GORE)

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



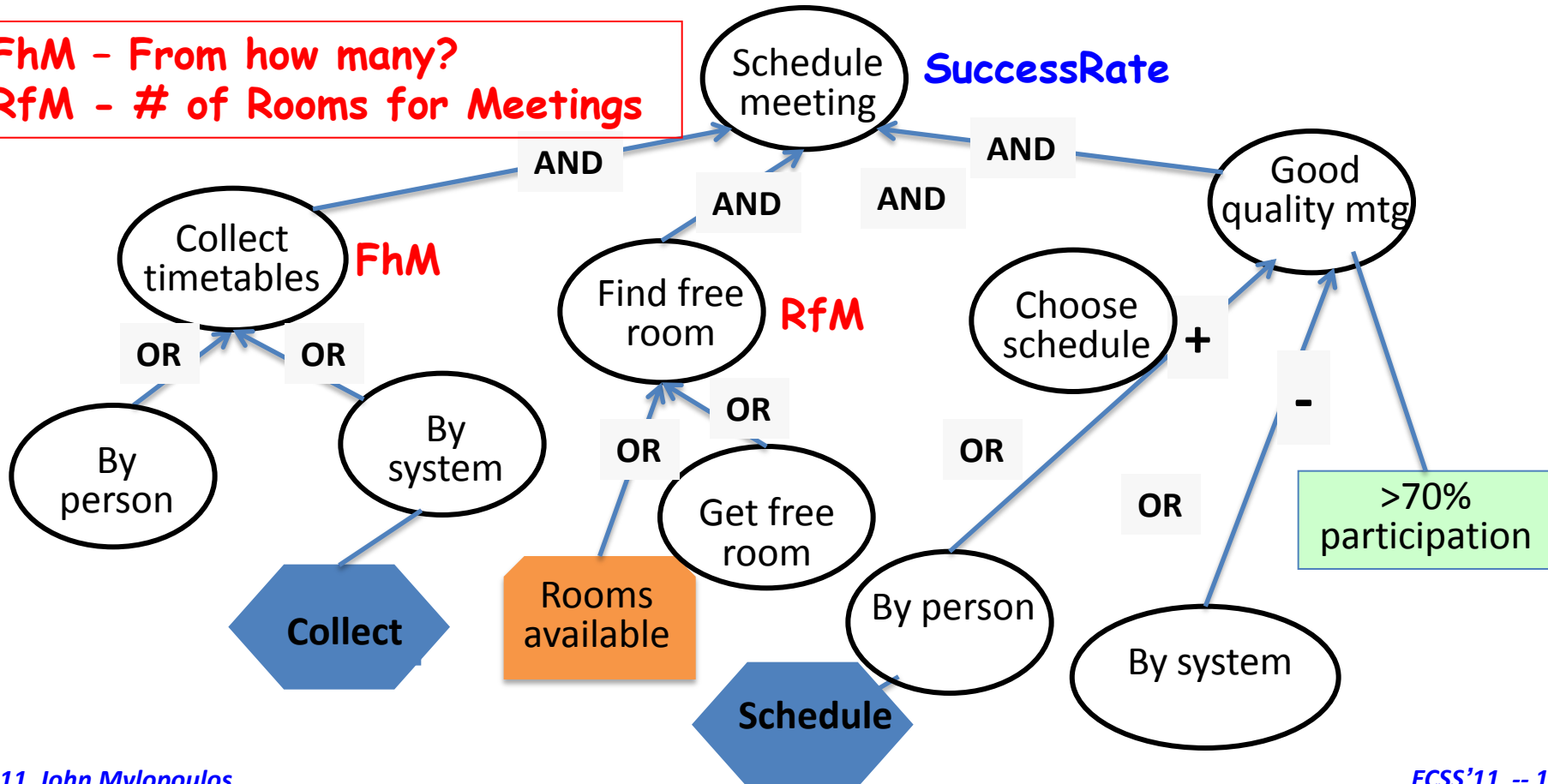
Goal problem spaces

- 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 (problem space)* and the requirements problem amounts to finding those that satisfy R, assuming D.
- 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.

Control variables and indicators (CV&I)

Success of a behaviour to fulfill its requirements depends on **control variables** and degree of success is measured by **indicators** (gauge/quality variables).

FhM - From how many?
RfM - # of Rooms for Meetings



The CV&I version of the RP

- Control variables and indicators determine resp. resource allocation, quality of performance.
- The requirements problem (CV&I) consists now of finding a solution that maximizes/does well enough with respect to one or multiple indicators.
- Now we have an optimization problem that seems to lie somewhere between OR-style optimization and SAT solving ...
- See [Letier11] for work along this direction.

The CV&I version of the RP

- Control variables and indicators determine resp. resource allocation, quality of performance.
- The requirements problem (CV&I) consists now of finding a solution that maximizes/does well enough with respect to one or multiple indicators.
- Now we have an optimization problem that seems to lie somewhere between OR-style optimization and SAT solving ...
- See [Letier11] for work along this direction.

Controlling CV&I software systems

🌈 To control a CV&I system, we need to add information about how control variables influence indicators

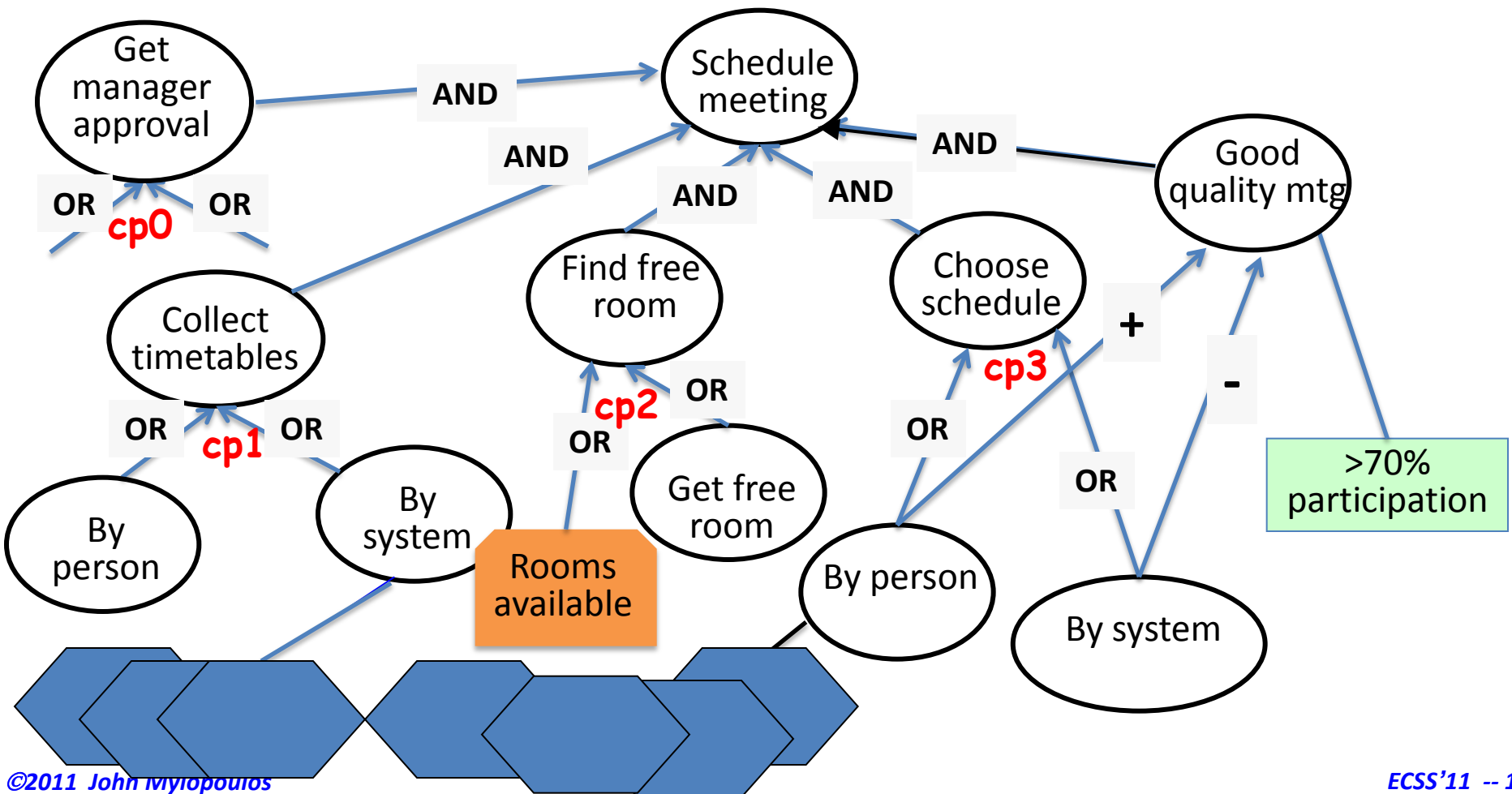
$$\text{e.g., } F(\text{FhM}, \text{RfM}, \text{SuccessRate}, \text{CostPerMtg}) = 0$$

$$G(\text{FhM}, \text{RfM}, \text{SuccessRate}, \text{CostPerMtg}) \geq 0$$

🌈 Such constraints -- in the form of differential equations and the like – constitute the backbone of Control Theory.

Dealing with unknown unknowns (MT)

🌈 Suppose now we have implemented several specifications and a running (=old) solution, and a requirement changes ...



The incremental requirements problem

- 🌈 All we need to do is run our GORE/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.

II. Research funding infrastructures

- Context: Recent move from Canada to Europe (Italy), after 39 years of academic research activity in Canada.
- Canadian research funding model (Science&Engineering)
- Funding programmes
 - ✓ Basic research: About 50% (*) of proposals funded, with small grants (€15K-75K/yr), no overhead applied.
 - ✓ Industry-University funding has different models.
- Features:
 - ✓ Inclusiveness
 - ✓ Continuity in funding
 - ✓ Stability of programmes, rules
 - ✓ Low overhead (researcher time, decision period, ...)
 - ✓ Strictly peer-reviewing system

European realities

- Funding of basic research (+)
- Inclusiveness (-/+)
- Continuity (-)
- Stability (--) [after factoring Italian context (--*)]
- Overhead (--), for application, administration and deliverables.
- Peer reviewing (+/-), factors other than peer-reviewing play an important role in some funding decisions.

Conclusions

- There is room for streamlining/rationalizing research funding in the EU.
- There is also room for more emphasis on
 - ✓ Basic research
 - ✓ Core CS areas, as opposed to fringe/emerging ones
 - ✓ Better interfaces with national funding programmes and rules

... In summary

- We have sketched a requirements-based approach to the design of software evolution mechanisms.
- We have also taken stock of EU research funding infrastructures, with Canadian glasses on ...



References

- 🌈 [Ernst11] Ernst N., Borgida A., Jureta I., “Finding Incremental Solutions for Evolving Requirements”, 19th Int. IEEE Requirements Engineering Conference (RE’11), Trento Italy, September 2011.
- 🌈 [Feather98] Feather M., Fickas S., van Lamsweerde A., Pounsard A., “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.
- 🌈 [Ingolfo11] Ingolfo S., Siena A., Mylopoulos J., “Establishing Regulatory Compliance for Software Requirements”, 30th International Conference on Conceptual Modelling (ER’11), Brussels, November 2011.
- 🌈 [Jackson95] Jackson M., Zave P., “Deriving Specifications from Requirements: An Example”, 17th International Conference on Software Engineering (ICSE’95).
- 🌈 [Letier11] Heaven W. and Letier E., “Simulating and Optimising Design Decisions in Quantitative Goal Models”, 19th Int. IEEE Requirements Engineering Conference (RE’11), Trento Italy, September 2011.

References (cont'd)

- [Lehman80] Lehman M., “On Understanding Laws, Evolution, and Conservation in the Large Program Life Cycle”, *Journal of Systems and Software* 1, 1980, 213–221.
- [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.
- [Siena10] Siena A., *Engineering Law-Compliant Requirements: The Nomos Framework*, PhD dissertation, Dept. of Inf. Eng. and Computer Science (DISI), University of Trento, 2010.
- [Souza11] Souza V., Lapouchnian A., Mylopoulos J., “The System Identification Problem for Adaptive Software Systems”, 30th International Conference on Conceptual Modeling (ER'11), Brussels, Nov. 2010.
- [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.
- [Yu10] Yu E., Giorgini P., Maiden N., Mylopoulos J. (eds.) *Social Modeling for Requirements Engineering*, MIT Press, ISBN: 978-0-262-24055-0, December 2010.

References (cont'd)

- [Lehman80] Lehman, M., “On Understanding Laws, Evolution, and Conservation in the Large Program Life Cycle”, *Journal of Systems and Software* 1, 1980, 213–221.
- [Liaskos10] Liaskos, S., McIlraith, S., Sohrabi, S., Mylopoulos, J., “Integrating Preferences into Goal Models for Requirements Engineering”, 18th 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.
- [Siena10] Siena A., *Engineering Law-Compliant Requirements: The Nomos Framework*, PhD dissertation, Dept. of Information Engineering and Computer Science, University of Trento, 2010.
- [Souza11] Souza V., Lapouchnian A., Mylopoulos J., “The System Identification Problem for Adaptive Software Systems”, 30th International Conference on Conceptual Modeling (ER’11), Brussels, Nov. 2010.
- [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.
- [Yu10] Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J. (eds.) *Social Modeling for Requirements Engineering*, MIT Press, ISBN: 978-0-262-24055-0, December 2010.