Requirements Models at Design- and Runtime

Alex Borgida
Rutgers University

John Mylopoulos (Trento,Toronto)

Fabiano Dalpiaz (Toronto)

Jennifer Horkoff (Trento)

Part of larger work on adaptive systems.

[Chidi Okoye]

OUTLINE

- General thoughts about models, modeling,...
- Specific Thoughts about Early Requirements Engineering
- Goal Oriented Requirements Modeling
  - Design time models
  - Run-time models

Warning: first part may be tendentious, opinionated; second part is work-in-progress

MODELS & MODELING

- "Models abstract away details for a purpose" (*wide agreement here yesterday)*
- **Many kinds of details**. [And to build big models you need to *refine* your original small model.]
  - classification [instantiation] (+in Taxis,RML:design [run-time])
  - generalization [specialization]
  - aggregation [decomposition]
  - "static" [dynamic/behavior]
  - specification [implementation]

(The above thoughts inspired by the field of Conceptual Modeling)

MODELING LANGUAGES/NOTATIONS/...

- *"models are for purpose":* what is "purpose"?
  - for us: answer questions
    - others (simulate/execute,...)
  - need a *semantics* on which to base |=
- language issues: i) expressiveness; ii) **ontology:** *what kinds of things are in the subject domain?*
  - this strongly influences the models you build by directing the kinds of questions you ask – be aware of it! (ER, Statecharts, FOPC (vs. Z),...)

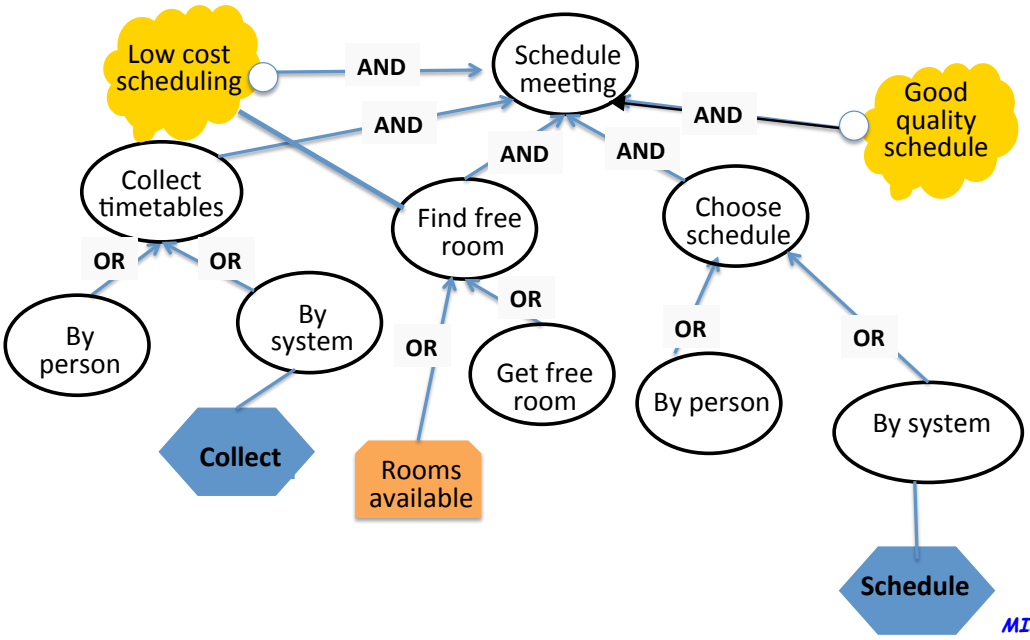
Requirements Engineering

- 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 (early) stakeholder requirements are necessarily *vague*, *informal*, *self-contradictory*, and more (... in short, "scruffy"), but they are requirements none-the-less!

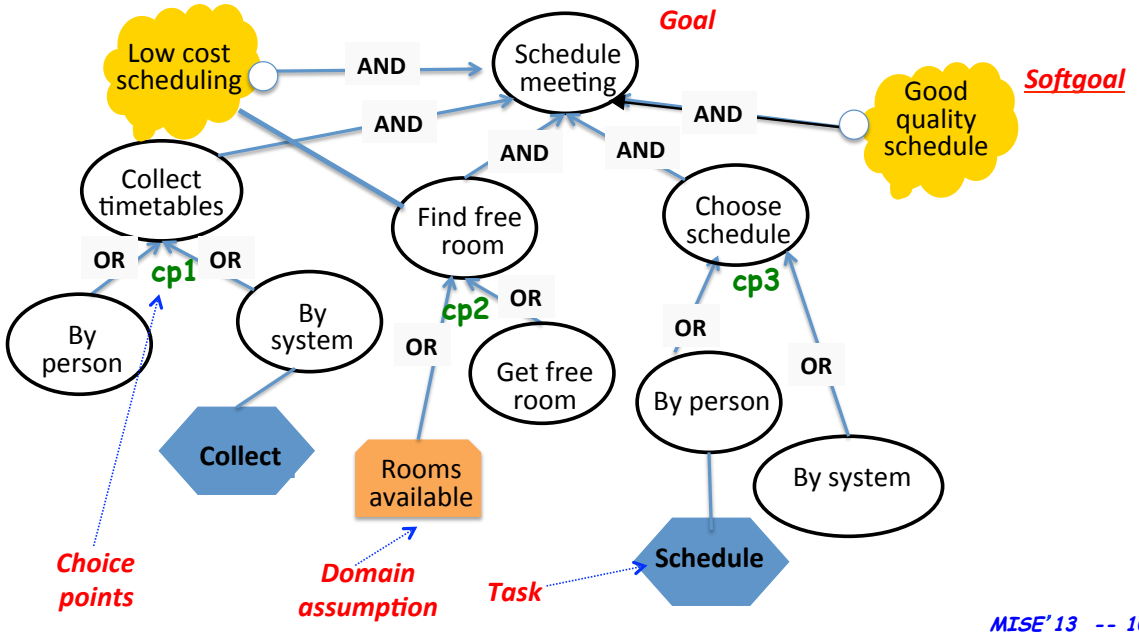
Interesting ideas

- Requirements derived from models of the ***domain*** (Ross).
- Requirements and ***specifications*** are different things, though logically related (Jackson).
- Requirements as ***goals*** stakeholders want (vanLamsweerde).
- The requirements problem is a ***social*** problem (Yu).
- The requirements problem is solved through problem ***refinement*** (all), and this refinement has many forms: activity decomposition (Ross), abductive inference (Jackson), goal refinement (van Lamsweerde), social delegation (Yu).
- With goal models and refinement, you are not exploring a design, but rather a design ***space***.

GORE: Goal Oriented Requirements Models

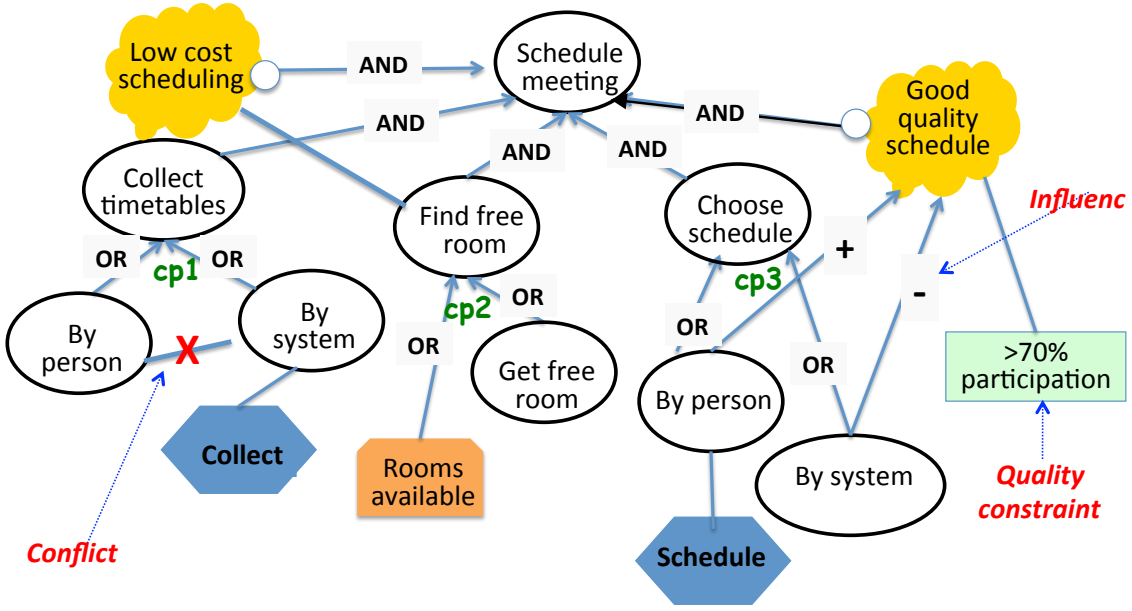


GORE Ontology



Goal Models circa 2013

- Goals can be mandatory/nice-to-have, can have priorities [Jureta08], probabilities [Letier04], utilities [Liaskos13], …



Reasoning with (design-time) Goal Models

- **What-if**: Assuming that some goals succeed/fail, infer the status of the rest of the goal model.
- **Satisfiability**: Is there a set of task specifications that achieve the top-level goals

(

- *What-if* reasoning can be handled with simple label propagation algorithms, satisfiability requires a min-SAT solver.
- Reasoning with preferences, probabilities and utilities requires more, e.g., AI planners [Liaskos10], SMT solvers, …

)

What do these models tell us?

- They give us alternative specifications (sets of functions qualities and assumptions) for fulfilling requirements.
- If someone wants a design that fulfills requirements in multiple ways (e.g., product families, flexible business processes, adaptive software systems) then our solution and implementation should encompass multiple specifications, not just one.
- These are ***design-time*** goal models, of no use during runtime and/or evolution.

Adaptive Software Systems

- Software systems increasingly operate within volatile environments where the one constant is *uncertainty*: cyber-physical systems, socio-technical systems, …
- In response, there has been growing interest in adaptive software that monitors its own performance and the environment, and adapts if its requirements fail.

⇨Need to monitor requirements, but how?

- Two approaches: (a) Monitor design artifacts (code, architecture, business process) and draw conclusions about requirements; (b) monitor requirements.

Design-time vs runtime models

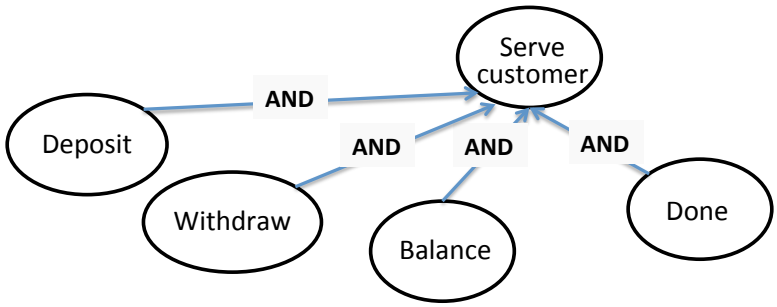
- Design-time models are intended to help us capture ***required functionality*** for the system-to-be.
- Runtime models are intended to help us ***monitor behaviour*** of the system and take corrective action, if necessary.

Runtime goal models

- Are augmented goal models that capture, in addition to a problem space
  - ✓ **Behaviour** – possible sequences of actions for fulfilling a goal;
  - ✓ **State** – possible states of a goal instance; current state of a goal instance;
  - ✓ **History** – the state history of all instances of a goal

Example

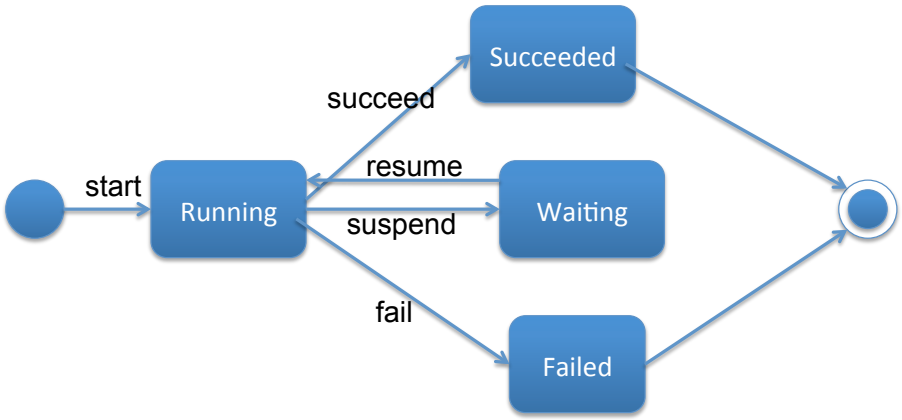
- Excerpt from example ATM model [Yiqiao Wang07]



- What questions might we ask about a runtime model? For example, if we know that for one instance of W, followed by 2 instances of D, all satisfied, and 2 instance of B, one satisfied, the other still being pursued, what is the state of the correspoding instance of SC?

State (fixed model)

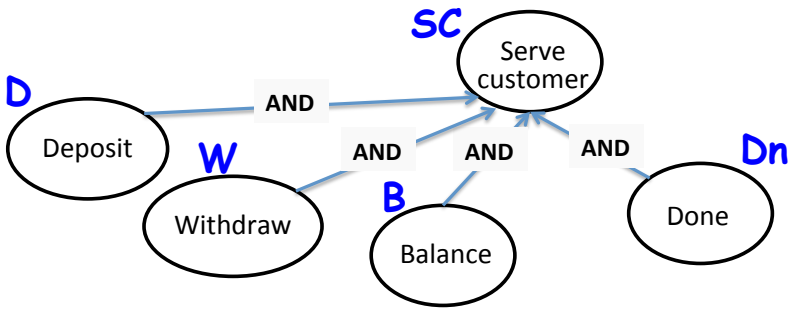
- We use FSMs, such as the following one (for goals).
- Every goal instance can be in one of these states. …



Behaviour: refining design time GM

- Described by annotating every non-leaf goal with a regular expression, e.g.,

  annot(SC) = (D | W | B)$^+$; Dn



Behaviour – Shuffle

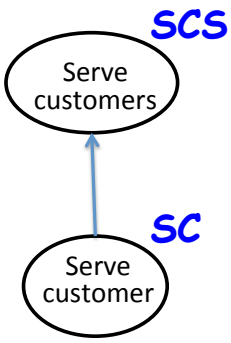
… or … ((D | W)$^+$ # B$^+$)

Means exactly what you expect …

- If w1 = abb and w2 = acbb,
  then w1 # w2 consists of strings
  like aabcbbb, aacbbbb, … lot's of them!
- More interestingly, shuffle closure …

  $w^{\#}$ = w | w#w | w#w#w | …

  allows for unbounded concurrency
- For example, annot(SCS) = $SC^{\#}$
- (Recognition for shuffle regular expressions is PTIME (in size of input trace only))

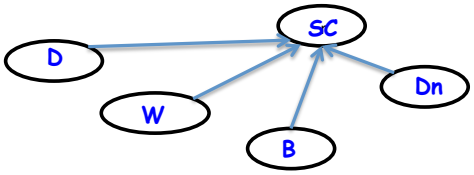


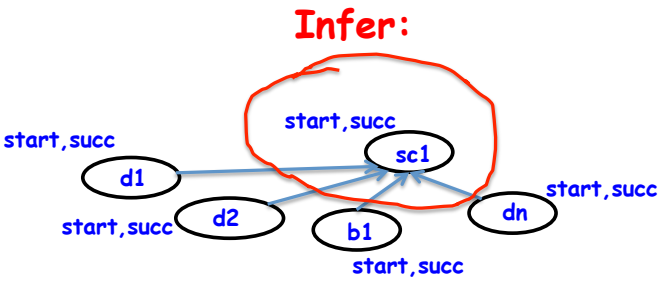
RUNTIME History

- At runtime, goals/tasks are instantiated (possibly many times)
- a system ***trace*** is a history of state transitions/events for goal instances:

d1.start,d1.succ,d2.start,…,dn.start,dn.succ

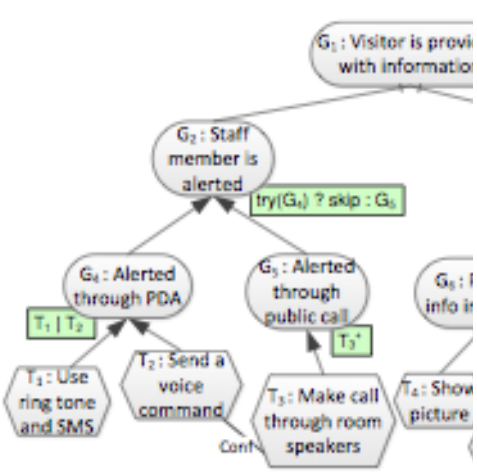
- REASONING: Also given DGM

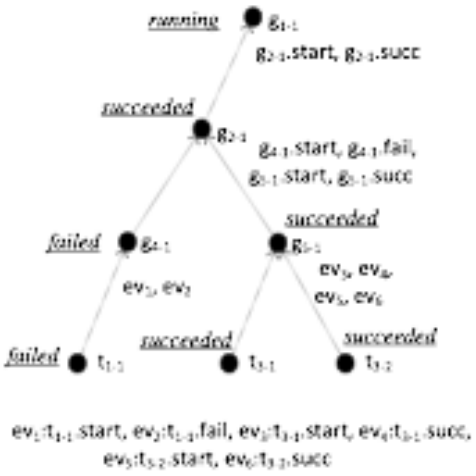




RUNTIME GOAL MODEL INSTANCE

- More general: *RGI* reconstruction from *partial (initial)* trace of ***leaf*** of instances





RUNTIME GOAL MODEL INSTANCE

- Consistency requirements
  - **children goal instances must satisfy behavioral annotation**
  - **states of children need to correlate with states of parents via RegExp rules** (e.g., if a step *fails,* supergoal *fails)*

Reasoning with Runtime Goal Models

- **Recognition**: Given a trace and a DGM, determine if the (partial) trace is legal.
- **RGI reconstruction**: Given a trace and a DGM, construct a corresponding goal instance model and infer the states of non-observable goal instances.
- **Diagnosis**: Assume a class of possible failures; given a trace and a DGM, determine if there is a failure; if so, determine all possible root causes.

Summary

- Unlike their design cousins, runtime requirements models need to capture behaviour, state and history.
- Reasoning for such models is founded on recognition problems for formal languages, rather than satisfiability.
- The ever-growing demand for flexibility, adaptability, customizability, etc. dictates the use of requirements models both at design time, runtime and throughout the lifecycle of a software system.

References

- [Borgida13] Borgida A., Dalpiaz F., Horkoff J., Mylopoulos J., "Requirements Models for Design- and Runtime", 2nd ICSE Workshop on Models in Software Engineering (MISE'13), San Francisco, May 2013.
- [Dalpiaz13] Dalpiaz F., Borgida A., Horkoff J., Mylopoulos J., "Runtime Goal Models", 7th IEEE International Conference on Research Challenges in Information Science (RCIS'13), Paris, May 2013.
- [Dardenne93] Dardenne, A., van Lamsweerde, A. and Fickas, S.,"Goal-Directed Requirements Acquisition", in The Science of Computer Programming 20, 1993.
- [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.
- [Letier04] Letier E., van Lamsweerde A., "Reasoning about Partial Goal Satisfaction for Requirements and Design Engineering", 12th International Symposium on the Foundation of Software Engineering FSE-04, pages 53–62, Newport Beach CA, November 2004.

References (cont'd)

- [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.
- [Liaskos13] Liaskos S., Khan S., Soutchanski M., Lapouchnian A., and Mylopoulos J., "Modeling and Reasoning About Uncertainty in Goal Models", (submitted for publication).
- [Ross77] Ross, D. T., and Schoman, "Structured Analysis: A Language for Communicating Ideas," *IEEE Transactions on Software Engineering 3(1)*, Special Issue on Requirements Analysis, January 1977, 16-34.
- [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] Silva Souza V., Lapouchnian A., Robinson W., Mylopoulos J., "Awareness Requirements for Adaptive Systems", 6th ICSE Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS'11), Waikiki Honolulu, May 2011.
- [Souza11a] Silva Souza V., Lapouchnian A., Mylopoulos J., "System Identification for Adaptive Software Systems: A Requirements Engineering Perspective", 30th International Conference on Conceptual Modelling (ER'11), Brussels, November 2011, 346-361.

References (cont'd)

- [Souza12] Silva Souza, V., Lapouchnian A., Angelopoulos K., Mylopoulos J., "Requirements-Driven Software Evolution", *Computer Science – Research and Development* (CSRD), Springer-Verlag, October 2012 (online version).
- [Souza12a] Souza V., Lapouchnian A., Mylopoulos J., "Requirements-Driven Qualitative Adaptation", 20th International Conference on Cooperative Information Systems (CoopIS'12), Rome, September 2012, Springer-Verlag LNCS 7566, 342-361.
- [Yu93] Yu Eric, "Modelling Organizations for Information Systems Requirements Engineering", First IEEE International Symposium on Requirements Engineering (ISRE'93), San Jose, January 1993.
- [Chidi Okoye] http://www.modernartimages.com/expressionsofdance3.htm