Simulating and Optimizing Design Decisions in Goal Models

Emmanuel Letier and William Heaven
Department of Computer Science
University College London
http://www.cs.ucl.ac.uk/people/E.Letier

Crest Open Workshop, London, 22 February 2011

Outline

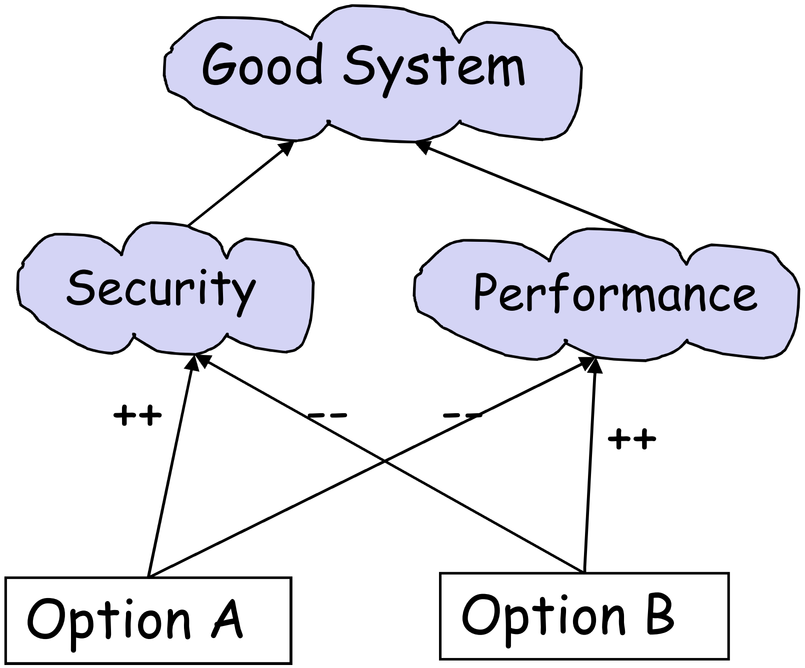
- Motivation
- Quantitative Goals Models
- Simulating and Optimizing Design Decisions

Context: Multi-Objective Design Decisions Problems

- Multiple stakeholders, multiple goals
  - Performance, usability, security, safety, cost, etc.
  - Goals are generally not directly comparable one to another
- Multiple design decisions
  - Which requirements to select for next release?
  - What component or actor will be responsible for what?
  - What actions to select to mitigate risks?

Qualitative Approaches

(NFR, Win-win, Soft System Methodology)

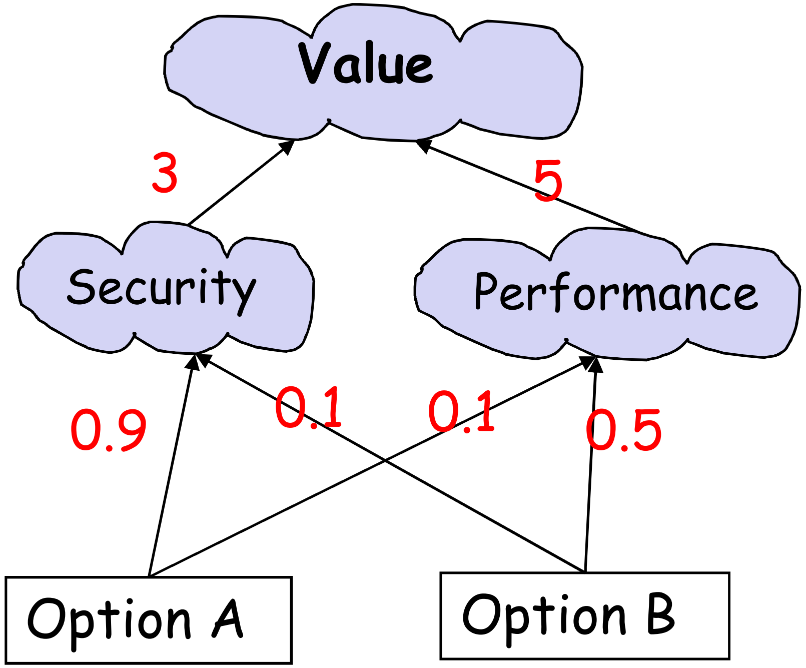


	Security	Performance
Option A	++	--
Option B	--	++

- ✓ Emphasis on understanding multiple perspectives and politics of the decision process, helps identifying hidden criteria
- ✗ Goal definitions remain too vague
- ✗ Qualitative information too poor to make informed decisions

Quantitative Approaches

(Cost-value based prioritization, Quality Function Deployment, NASA's DDP, etc.)



	Security	Performance	**Cost**
Option A	0.9	0.1	54
Option B	0.1	0.5	42

- Basically: replace ++ and –- by numbers, count the cost
- Compute overall value, usually as weighted average

~~The Big Question~~

The Wrong Question

Where do the numbers come from?

The Good Questions

- *What do the numbers mean?*
- *How do we know they are correct?*

What do the numbers mean?

How do we know they are correct?

- Predicted objective attainments

E.g. Value = 76 ; cost = 42

Security = 7.88 ; Performance = 9.57

?

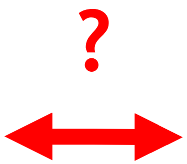
What do the numbers mean?

How do we know they are correct?

- Equations used to compute objectives from parameters

E.g. $Cost = \Sigma\, selected(req_i) * cost(req_i)$

$Value = \Sigma\, weight(G_k) * satisf(G_k)$



What do the numbers mean?

How do we know they are correct?

- Model parameters

E.g. Cost(req_i) = 3; contrib(req_i, Security) = 0.5

weight(Security) = 3 ; weight(Performance) = 5

?

Two Fundamental Principles

- Requirements descriptions should be testable
- Early design decisions must deal with uncertainties

Quantitative Goal Models

E. Letier, A. van Lamsweerde,

Reasoning about partial goal satisfaction for requirements and design engineering,

FSE 2004

Specifying Levels Goal Satisfaction

- Quantitative goal = behaviour goal extended with objective functions defined in terms of domain quality variables (formally, random vars)

Goal Achieve [Ambulance Intervention]

Def. An ambulance must arrive at incident scene within 14 min. after the first call

Objective Functions

Max 14Min_RespRate = *P* (RespTime ≤ 14')

Max 8Min_RespRate = *P* (RespTime ≤ 8')

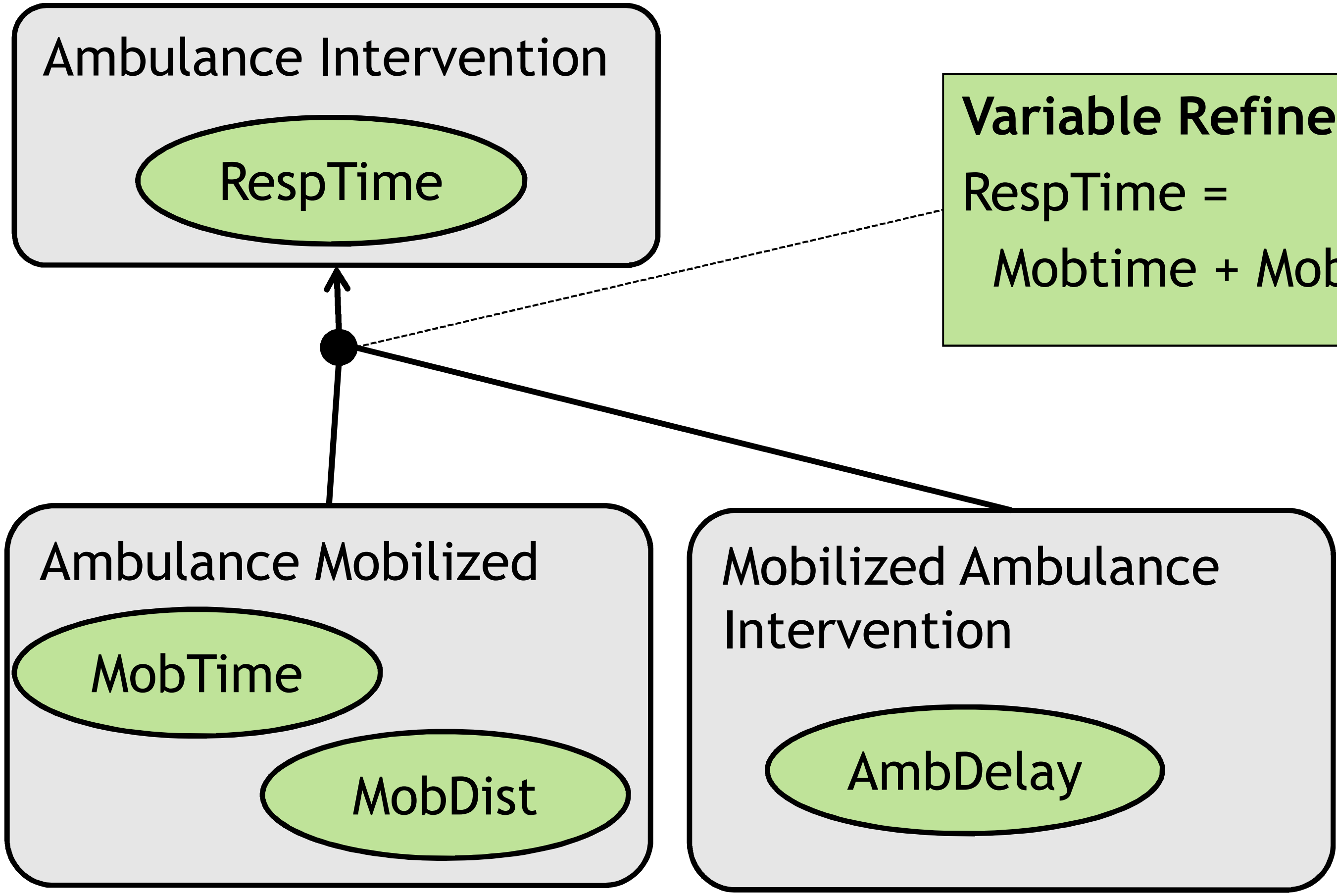
Quality Vars

RespTime: Incident -> Duration

- Based on industrial practices: VOLERE, Planguage, etc.

Specifying quality variables refinement equations

- Equations relating quality variables of parent goal to quality variables of subgoals (and related domain properties)



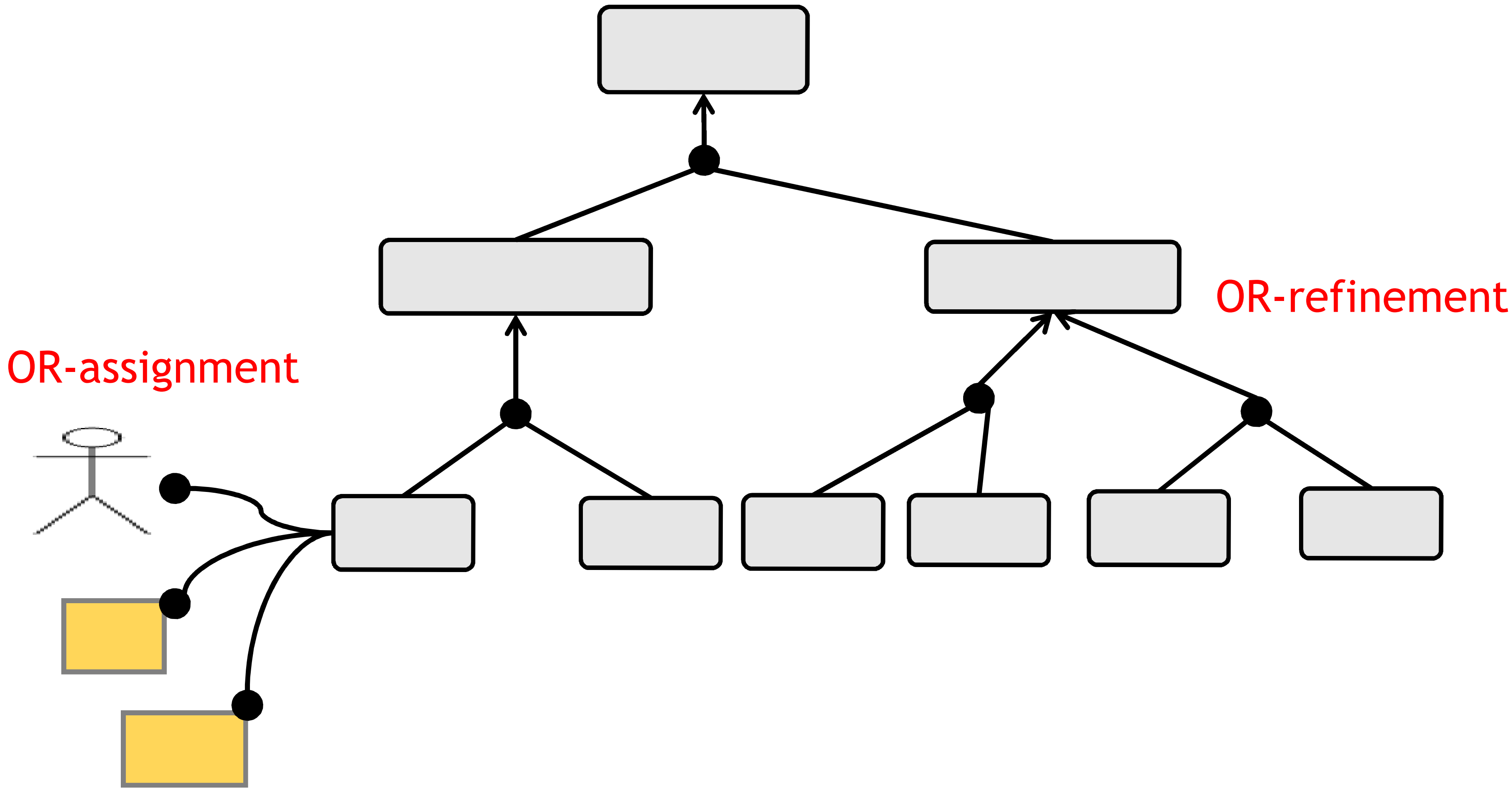
Variable Refinement:

RespTime =

Mobtime + MobDist+ AmbDelay

Modelling alternatives

- *Decision points* = goal refinements, responsibility assignments, obstacle resolutions



Estimating leaf quality variables

Variables Estimations:

AmbDelay = Normal(0, 120) % Normal distribution

CallTakingTime = Exp(150) % Exponential distribution

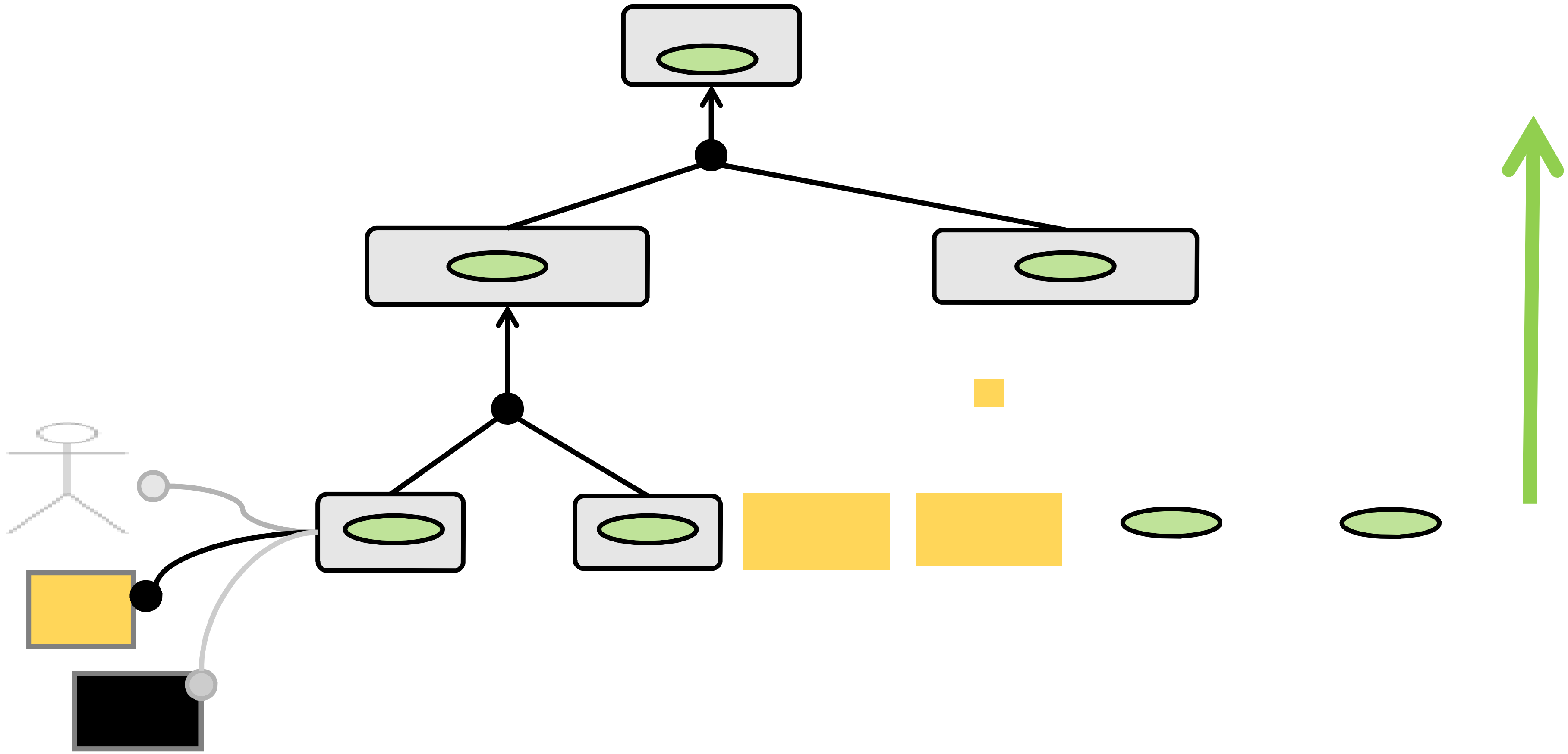
…

(Constraint: leaf quality variables must be statistically independent)

- Estimations can be *descriptive*, *predictive*, or *prescriptive*
- Ideally, all should be *testable*

Computing Objective Functions

… for one particular set of design decisions



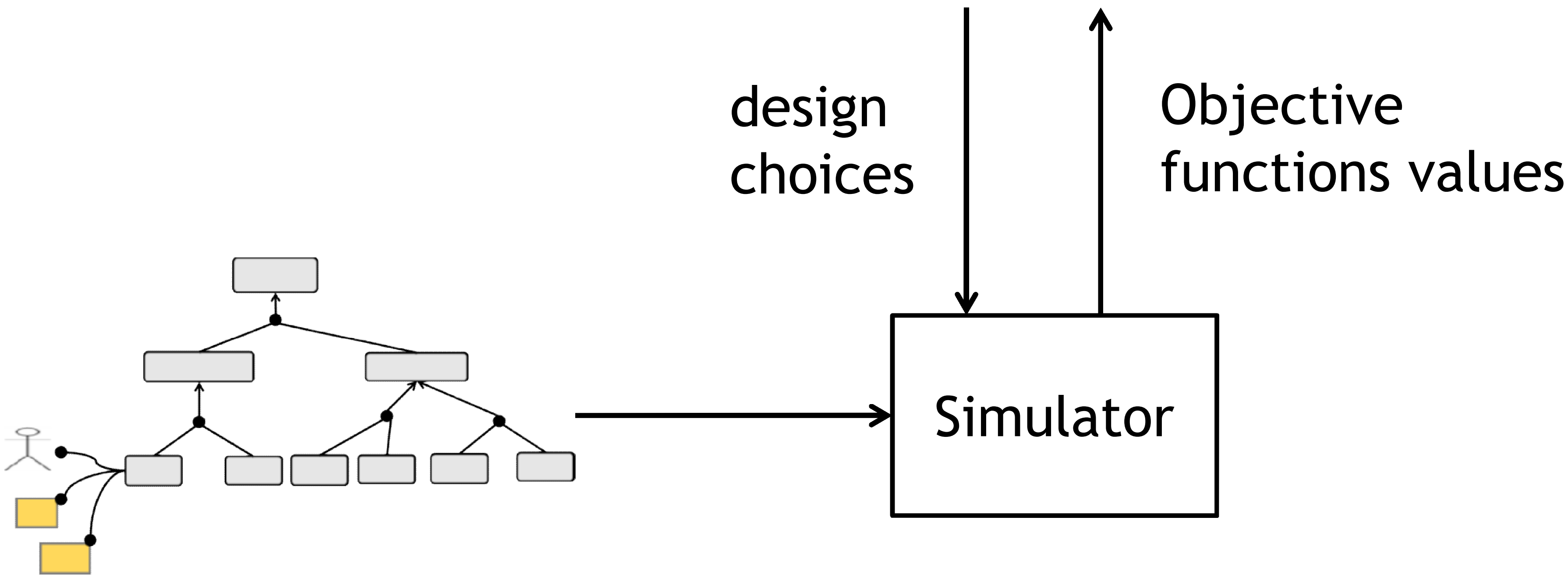
Complex because arbitrary equations and probability distributions

Simulating and Optimizing Design Decisions

With William Heaven, UCL

Simulating the model

- Compute objective functions through stochastic simulation



Goal Simulation

Function Simulate(Goal G)

Inputs

N_S size of sample space S for each quality variable

Outputs

E(Obj) estimated value for each objective fn. Obj

S(X) array of N_S simulated value for each quality var. X

Example

Simulate(Ambulance Intervention)

Input $N_{incident}$ = 1000

Output

E(14Min_RespTime) = 90%

E(8Min_RespTime) = 60%

S(RespTime) = [11’30”, 7’50” , 14’05”, 10’10”, …]

Simulation algorithm

Goal graph traversed recursively from top goals

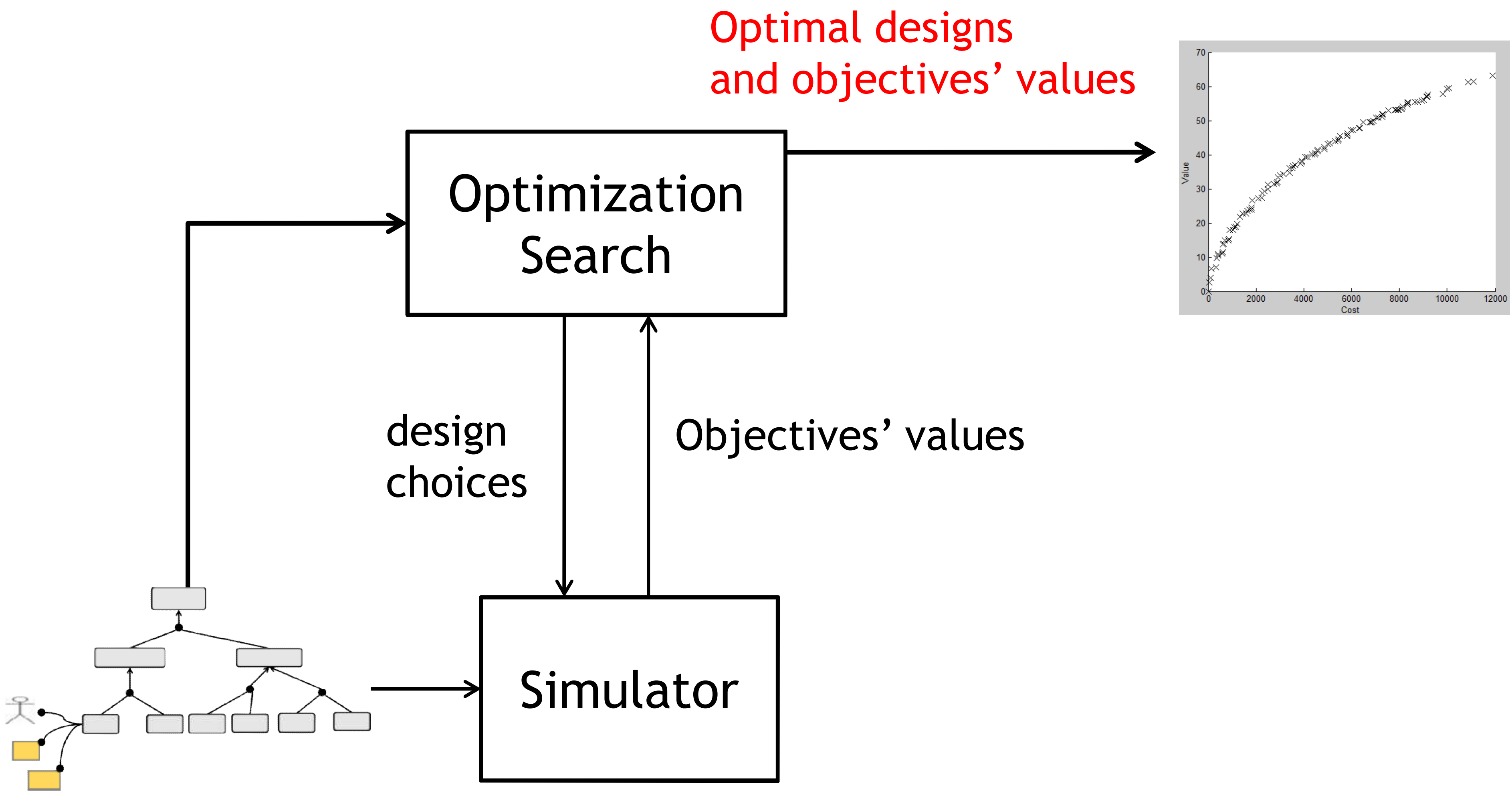
Simulate (G, [N_1, …, N_n])

←

1. Simulate each subgoal (if it has not been simulated before)
2. Compute G's quality variables arrays from subgoals quality variables and refinement equations
3. Compute G's objective functions from quality variable arrays

Prototype implementations in R and Matlab

Optimizing Design Decisions



Current and Future Projects

- Integrated tool support for simulation and optimization
  - Editor + fully connected components
  - Combining discrete and continuous design variables
  - Performance, grid computing
  - Sensitivity and robustness analysis
- Dealing with the optimization inputs
  - Model construction
  - Elicitation of model parameters and (meta-) uncertainties
- Dealing with the optimization outputs
  - Helping decision makers analysing sets of optimal solutions