



# Goal Orientation

Mariano Ceccato

[mariano.ceccato@univr.it](mailto:mariano.ceccato@univr.it)



# System objectives are pervasive in RE



As seen before ...

- the WHY dimension of RE (introduction lecture)
  - understanding objectives in system-as-is, eliciting objectives of system-to-be (elicitation lecture)
  - analyzing conflicting objectives, analyzing risks of not meeting critical objectives, evaluating options against objectives (evaluation lecture)
  - specifying the rationale for specific requirements (specification lecture)
  - checking that system objectives are satisfied by operational requirements (quality assurance lecture)
  - documenting satisfaction arguments & backward traceability to system objectives (evolution lecture)
- ⇒ **Goals** as key abstraction for driving the RE process



# Outline

- What are goals?
- The granularity of goals and their relationship to requirements and assumptions
- Goal types and categories
  - Types of goals: behavioral goals vs. soft goals
  - Goal categories: functional goals vs. non-functional goals
- The central role of goals in the RE process



# What are goals?

- Goal = prescriptive statement of intent the system should satisfy through cooperation of its agents
  - "prescriptive statement": in optative mood "shall", "should", "must", ...
    - e.g. "Train doors shall be closed while the train is moving"
    - "Loan periods shall be limited to 2 weeks"
- formulated in terms of problem world phenomena
- "system": system-as-is, system-to-be
- "agent": active system component responsible for goal satisfaction





# Goal satisfaction requires agent cooperation



Maintain [SafeTransportation] ↔

on-board train controller + tracking system + station computer + passenger +  
train driver + ...



Achieve [BookCopyReturnedToShelves] ↔

patron + staff + library software

- Agent = role, rather than individual
  - must restrict its behavior to meet its assigned goals
  - must be able to monitor/control phenomena involved in assigned goals
- Agent types
  - software (software-to-be, legacy software, foreign software)
  - device (sensor, actuator, ...)
  - human



# Goals vs. domain properties



- Domain property = descriptive statement about environment
  - indicative mood: "is", "are", etc --not prescriptive
  - e.g. "If train doors are open, they are not closed"  
"A borrowed book is not available for other patrons"
- The distinction between *goals* & *domain properties* is essential for RE
  - goals can be negotiated, weakened, prioritized
  - domain properties cannot
  - both required in requirements documentation



# The granularity of goals



- Goals can be stated at different levels of abstraction
  - **Higher-level** goals: strategic, coarse-grained
    - "50% increase of transportation capacity"
    - "Effective access to state of the art"
  - **Lower-level** goals: technical, fine-grained
    - "Acceleration command sent every 3 secs"
    - "Reminder issued by end of loan period if no return"
- The **finer**-grained a goal, the **fewer** agents required for its satisfaction





# Goals, requirements & expectations



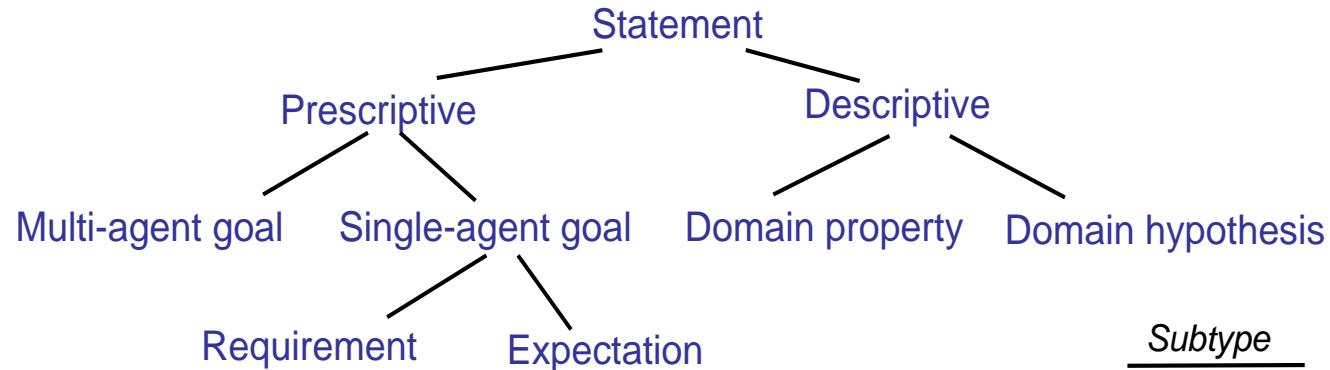
- **Requirement** = goal assigned to single agent in software-to-be  
"doorState = 'closed' while measuredSpeed ≠ 0" ↔ *TrainController*  
"Acceleration command sent every 3 secs" ↔ *StationComputer*
- **Expectation** = goal assigned to single agent in environment
  - prescriptive assumption on environment
  - cannot be enforced by software-to-be (unlike requirements)"Train left when doors open at destination" ↔ *Passenger*



# Statement typology revisited in the presence of goals

Cf. general terminology introduced in intro lecture ...

- software requirement  $\leftrightarrow$  requirement
- system requirement  $\leftrightarrow$  goal involving multiple agents incl. software-to-be
- (prescriptive) assumption  $\leftrightarrow$  expectation
- (descriptive) assumption  $\leftrightarrow$  hypothesis



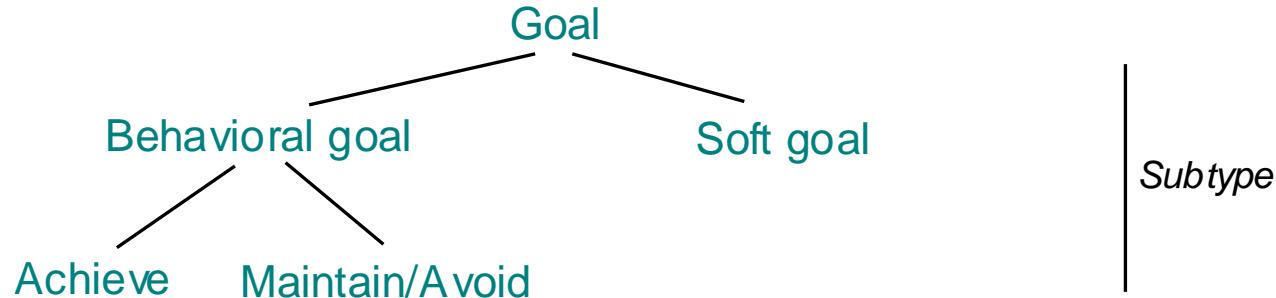


# Goal types

Behavioral goals: prescribe behaviors

vs.

Soft goals: state preferences among alternative behaviors





# Goal types: behavioral goals



- Prescribe intended system behaviors declaratively
  - implicitly define maximal sets of admissible agent behaviors
- Can be satisfied in a clear-cut sense: YES or NO
  - goal satisfaction, formal analysis
- Used for building operation models to meet them

"Worst-case stopping distance maintained"

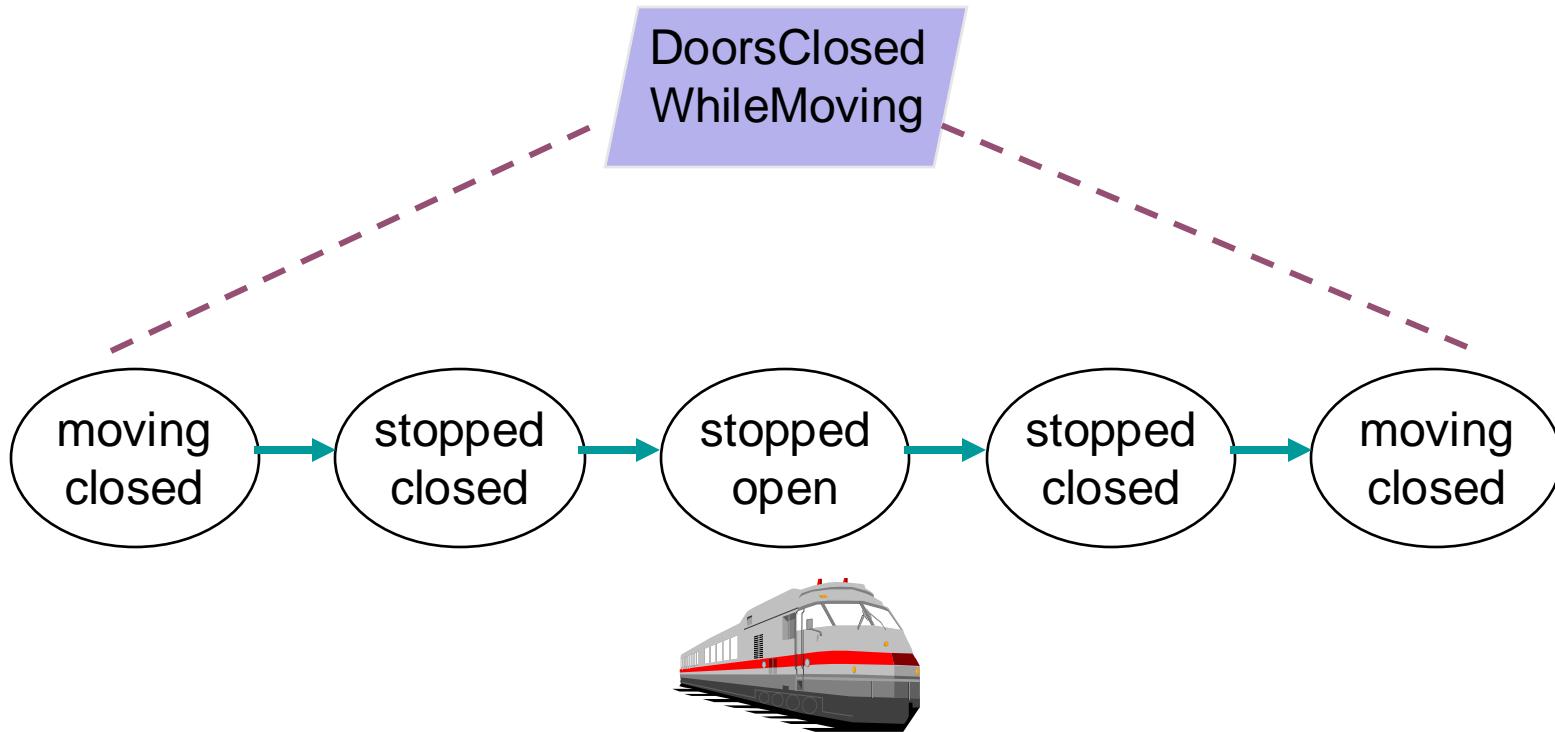


"Reminder sent if book not returned on time"





# Behavior goals prescribe sets of desired behaviors





# Behavioral goals: subtypes and specification patterns

- **Achieve** [TargetCondition]:

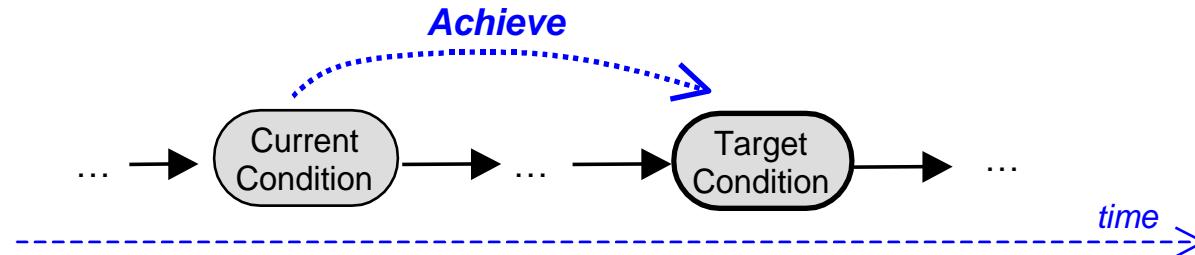
- [if CurrentCondition **then**] **sooner-or-later** TargetCondition

**Achieve** [BookRequestSatisfied]:

if a book is requested **then sooner-or-later**  
a copy of the book is borrowed by the requesting patron

**Achieve** [FastJourney]:

if train is at some platform **then within 5 minutes** it is at next platform





# Behavioral goals: subtypes and specification patterns

- **Maintain** [GoodCondition]:

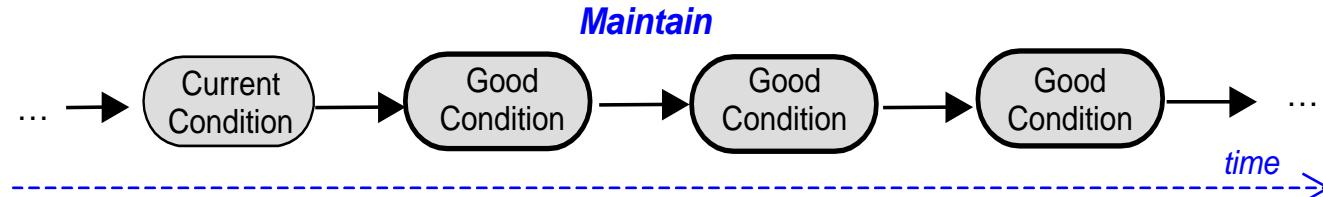
- [if CurrentCondition then] **always** GoodCondition
- **always** (if CurrentCondition then GoodCondition)

**Maintain** [DoorsClosedWhileMoving]:

**always** (if a train is moving then its doors are closed)

**Maintain** [WorstCaseStoppingDistance]:

**always** (if a train follows another then  
its distance is sufficient to allow the other to stop suddenly)





# Behavioral goals: subtypes and specification patterns

- Accuracy goals are usually of type *Maintain*  
*Maintain [AccurateBookClassification]:*  
    if a book is registered in the library directory **then**  
        **always** its keyword-based classification reflects its covered topics

- **Avoid** [BadCondition]: dual of *Maintain* ...
  - [if CurrentCondition **then**] **never** BadCondition

*Avoid [BorrowerLoansDisclosed]:*  
    **never** patron loans disclosed to other patrons

Many security goals are Avoid goals



# Goal types: soft goals

- Capture preferences among alternative behaviors
- Can not be satisfied in clear-cut sense:  
    **more** satisfied in one option, **less** satisfied in another
  - goal satisficing, qualitative analysis
- Used for comparing options to select preferred
- Often take the form  
*Maximize / Minimize, Increase / Reduce, Improve, ...*

“Stress conditions of air traffic controllers shall be reduced”

“The workload of library staff shall be reduced”

“The bibliographical search engine shall be usable by non-CS students”



# Goal categories

- Classification into **functional, quality, development** goals
- Categories may overlap; boundary not always clear
  - unlike goal types
- **Functional goals**
  - prescribe intended services to be provided by the system
  - used for building operational models of such services
    - use cases, state machines (see later)

“Passengers transported to their destination”

“Train acceleration computed”

"Book request satisfied"

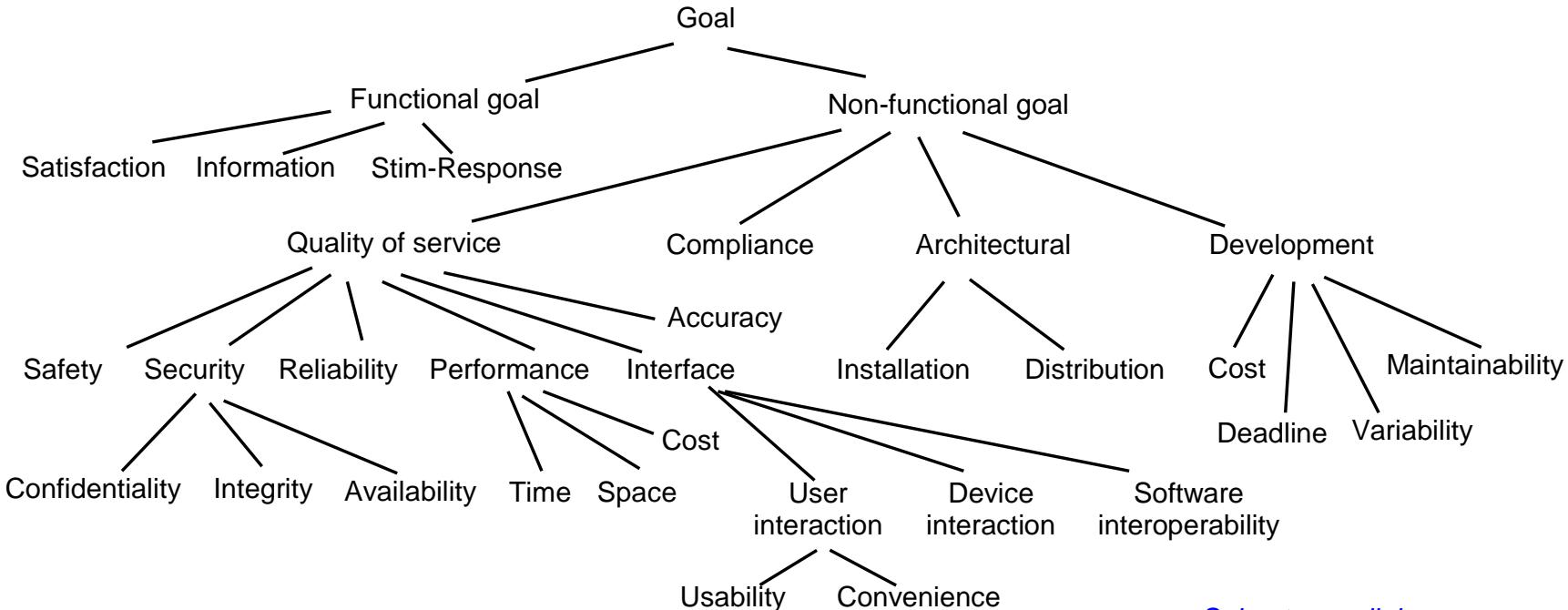


# Goal categories: non-functional goals

- **Quality goals** (not to be confused with soft goals)
  - about quality of service ...
    - security "info about other patrons kept confidential"
    - safety "worst-case stopping distance maintained"
    - accuracy "measured speed = physical speed"  
"book displayed as available iff there is a copy in shelves"
    - performance "acceleration command sent every 3 seconds"
    - usability
    - interoperability, ...
- **Development goals**
  - about quality of development ...
    - cost, deadline, variability, maintainability, reusability, etc.



# Goal categories



[Subcategory link](#)

Helpful for eliciting overlooked application-specific instances through taxonomy browsing



# Using goal types & categories

- Lightweight specification patterns
- Heuristic rules for elicitation, validation, reuse, conflict management,  
...

"Is there any conflict between **Information** goals and **Confidentiality** goals?"

"**Confidentiality** goals are **Avoid** goals on sensitive info"

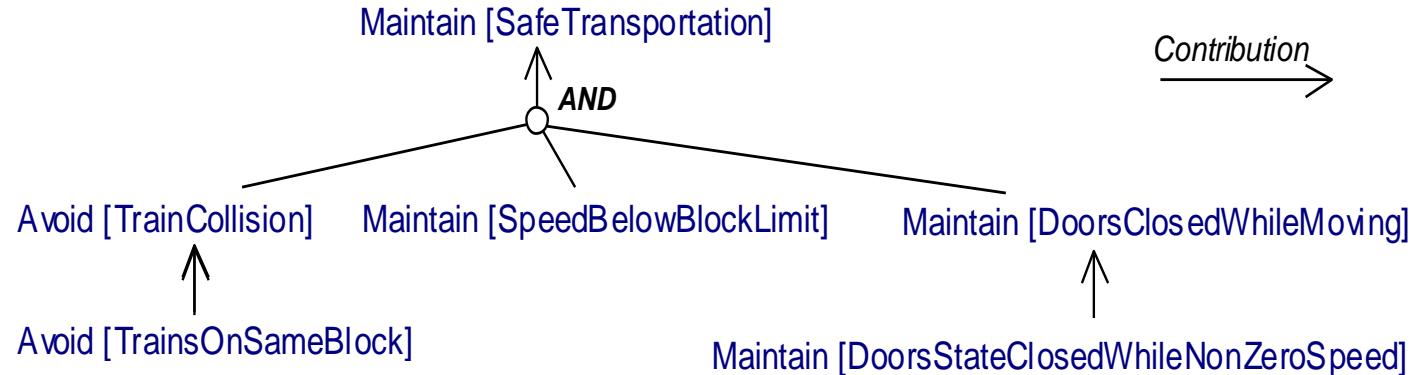
"**Safety** goals have **highest priority** in conflict resolution"

*More specific types & categories ⇒ more specific heuristics*



# The central role of goals in the RE process

- Goal refinement/abstraction as structuring mechanism
  - shows contribution links among goals
  - drives elaboration of reqs (subgoals)
  - provides rationale for reqs (parent goals)
  - rich traceability: strategic objectives → technical requirements
  - can be used to structure reqs document





# The central role of goals in the RE process

- Goals support chains of satisfaction arguments (cf. intro lecture)

**Req**, Exp, Dom |=  $G$ , SubG, Exp, Dom |=  $G$

“in view of domain properties in *Dom*, the reqs/subgoals in *Req/SubG* ensure that goal **G** is satisfied under expectations in *Exp*”

**R:** doorsState = 'closed' if measuredSpeed ≠ 0

**E:** Doors are closed iff doorsState = 'closed' ( $\leftrightarrow$  door actuators)

`measuredSpeed = physicalSpeed`      ( $\leftrightarrow$  speedometer)

**D:** Train is moving iff physicalSpeed ≠ 0

**G:** Doors are closed if train is moving



# The central role of goals in the RE process

- Goals provide a criterion for reqs **completeness**  
set REQ of requirements is complete if for all goals  $G$  :

$$\{\text{REQ}, \text{Exp}, \text{Dom}\} \models G$$

- Goals provide a criterion for reqs **relevance**

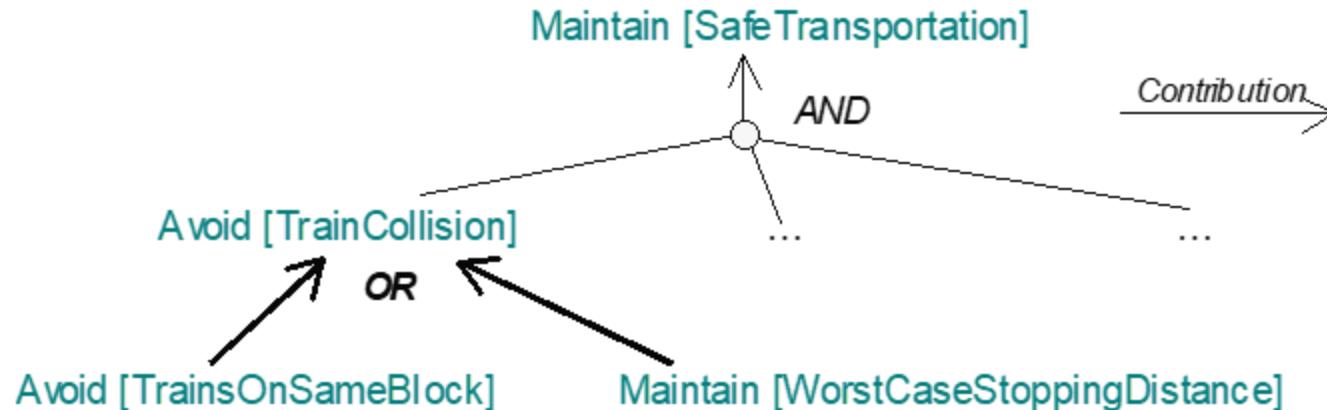
r in REQ is pertinent if for some  $G$  :

$$r \text{ is used in argument } \{\text{REQ}, \text{Exp}, \text{Dom}\} \models G$$



# The central role of goals in the RE process

- Goal OR-refinement → capture of alternative options





# The central role of goals in the RE process

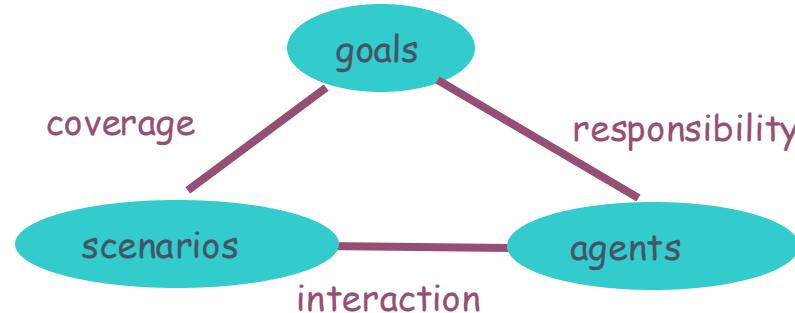
- Support for evolution management
  - higher-level goals → more stable concerns
  - ⇒ multiple system versions within single model ...
    - common parent goals
    - different OR-branches
- Roots for conflict detection & resolution
- Anchors for risk management



# Avoid frequent misconceptions

- Goal-oriented  $\neq$  top-down
  - bottom-up elaboration as well (goal abstraction)
- Goal-oriented  $\Rightarrow$  agent-oriented, scenario-oriented

*the magic RE triangle:*





# Scenarios as concrete vehicles for goal elicitation/validation

easy to get from or validate with stakeholders

