Software Engineering

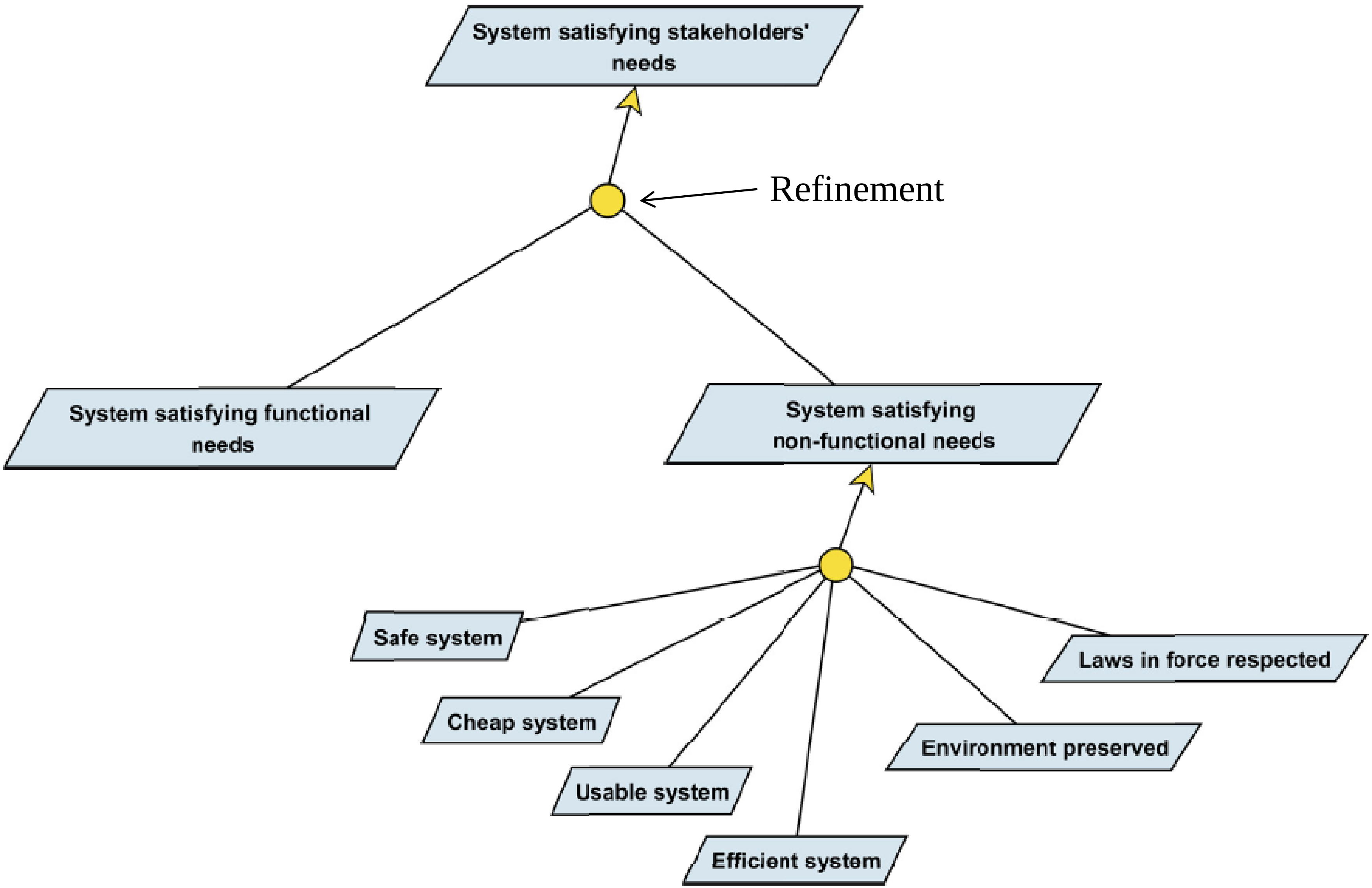
Goal-Oriented Requirements Engineering

The KAOS approach (cont.)

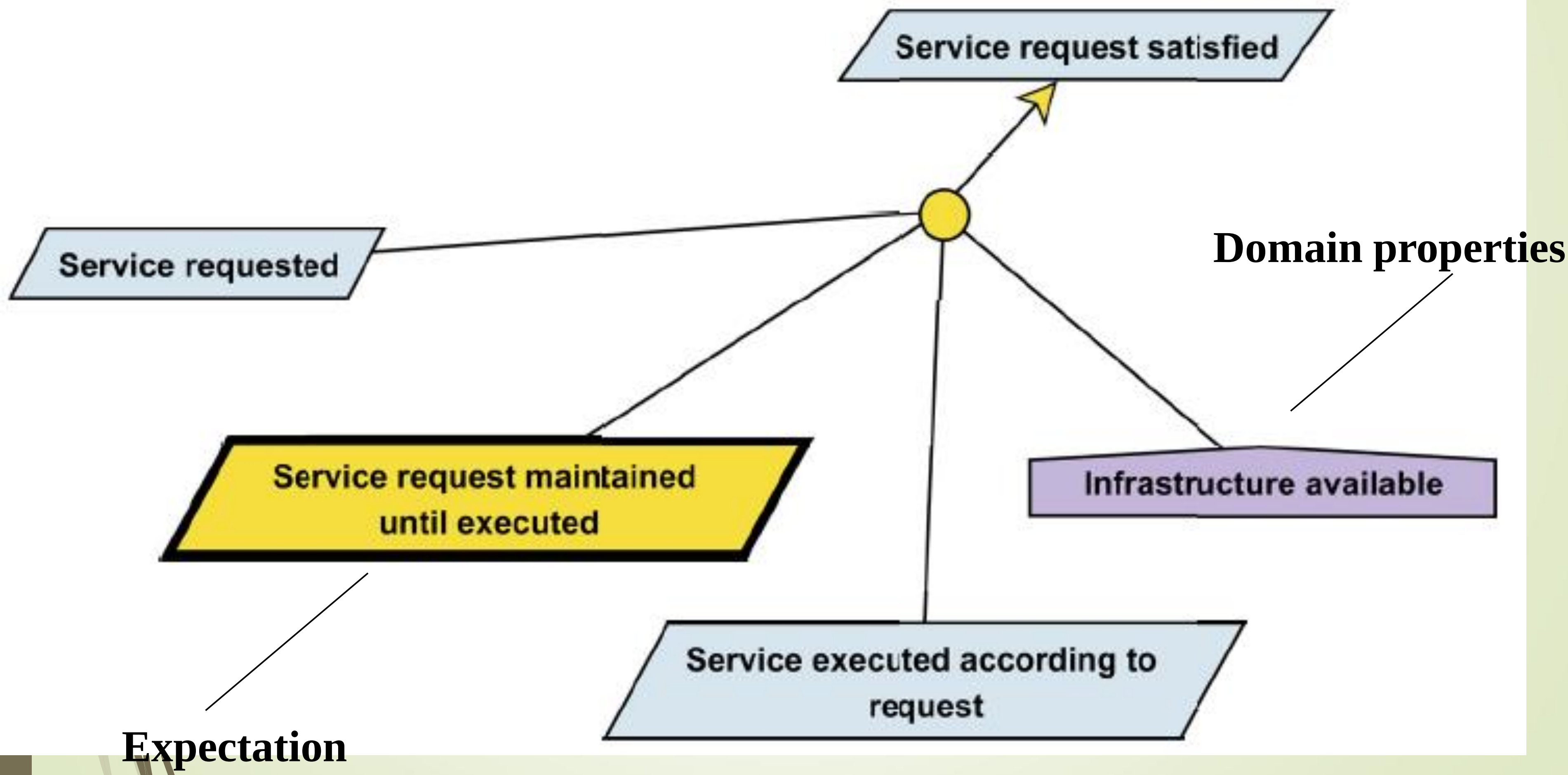
Goal Model and Patterns

- Requirements patterns
  - Reqs can be obtained through interviews or docs.
  - An efficient way is to reuse patterns
- KAOS consists of modelling generic patterns of reqs.
  - They are progressively built

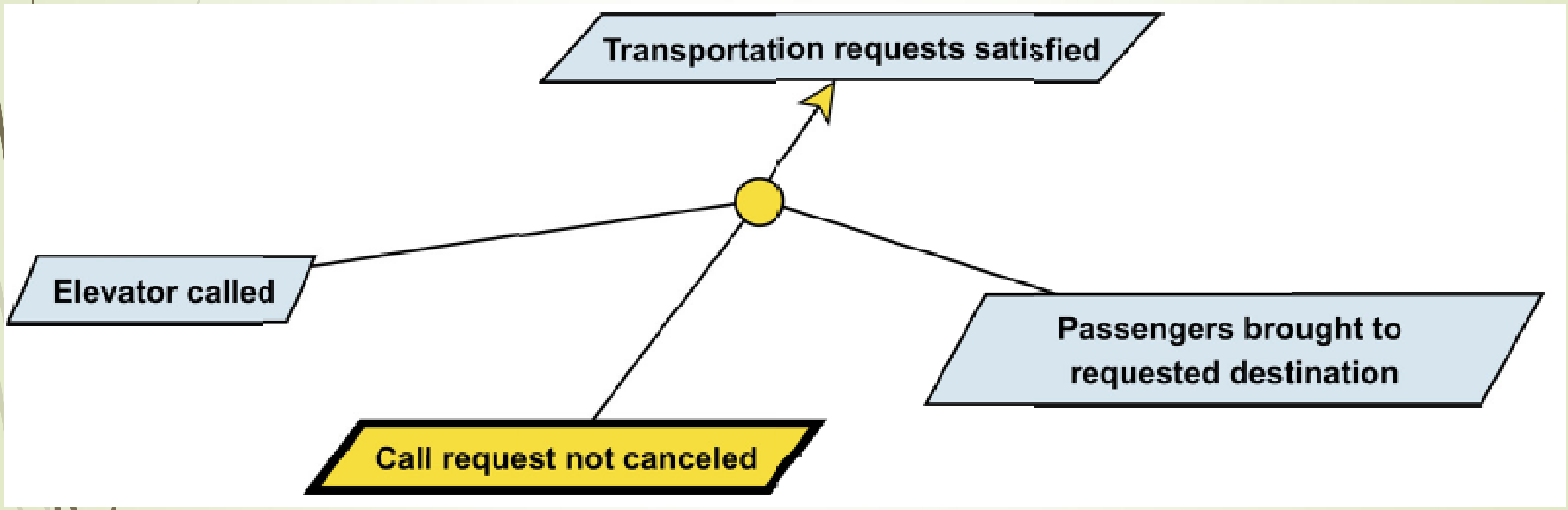
Generic Goal Pattern



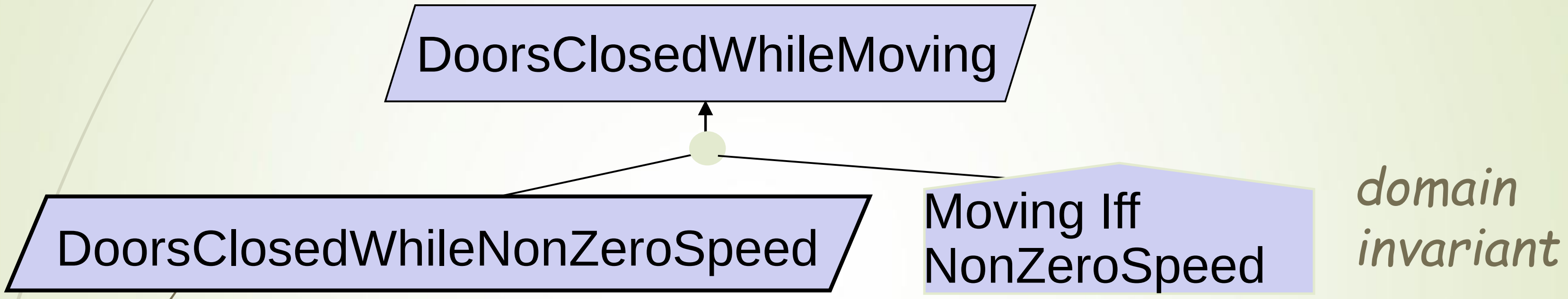
Generic Pattern: Expectations and Domain Properties



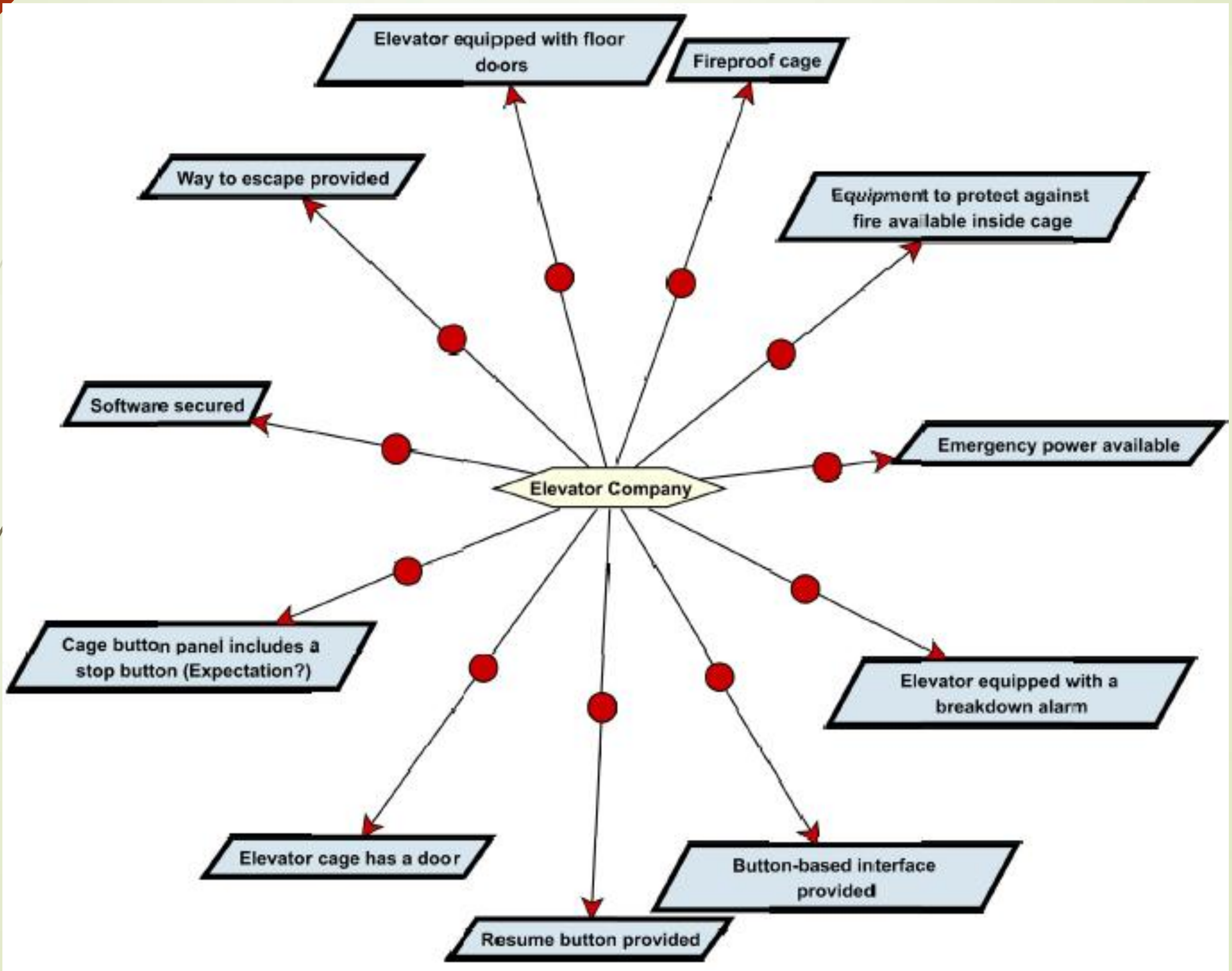
Application of the pattern to the elevator system



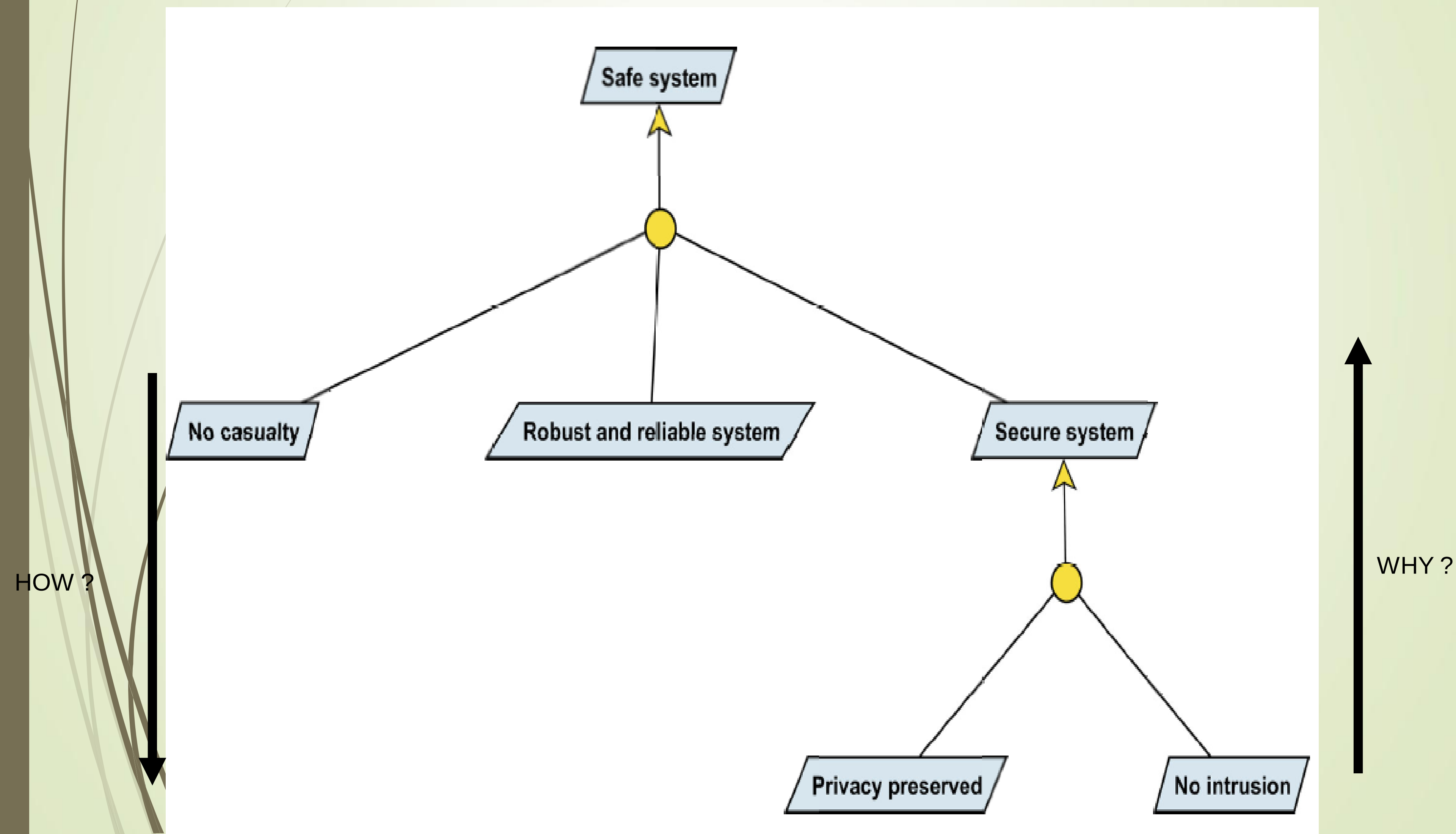
Domain properties in AND-refinements in the train example

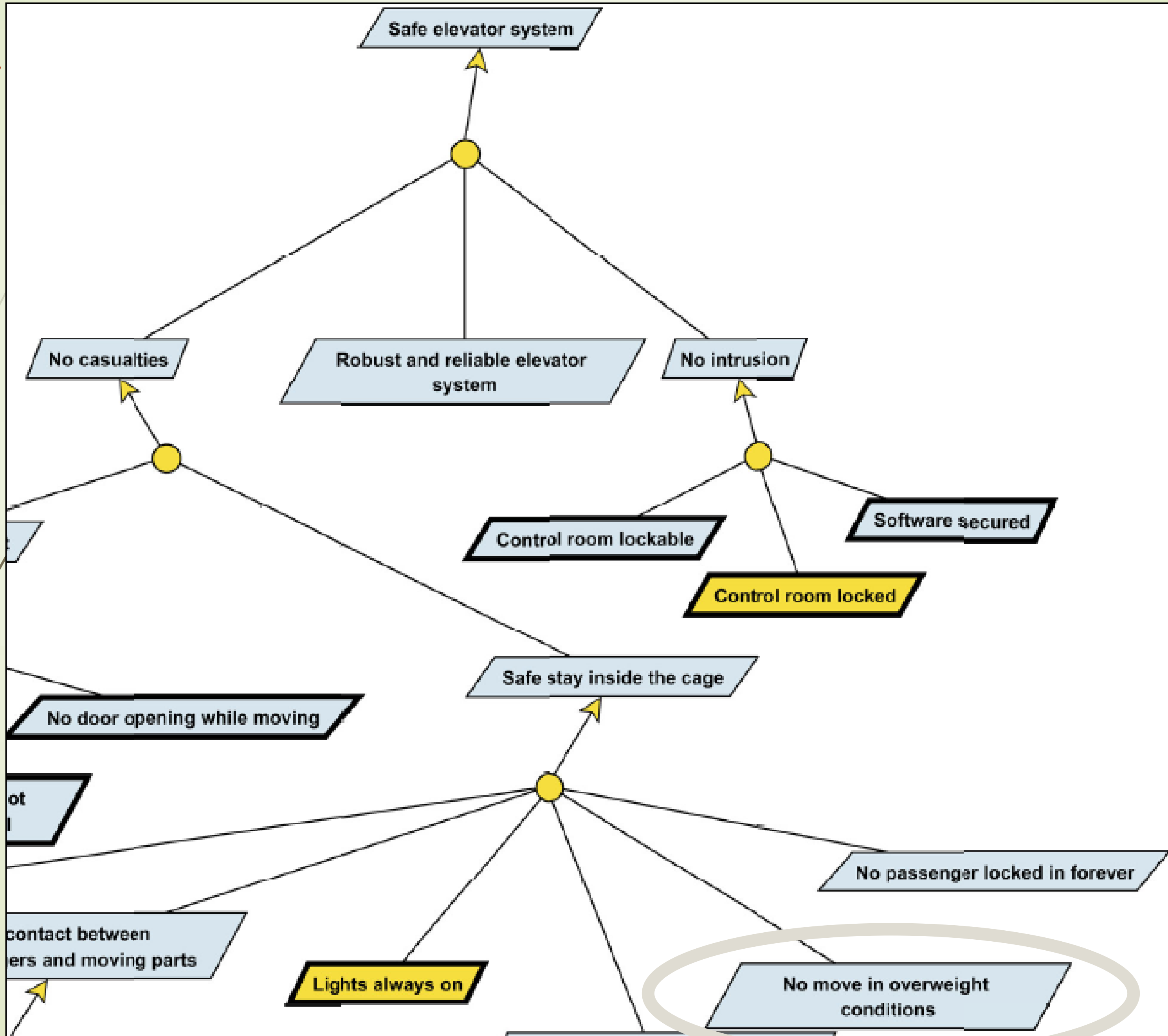


Responsibilities of an agent

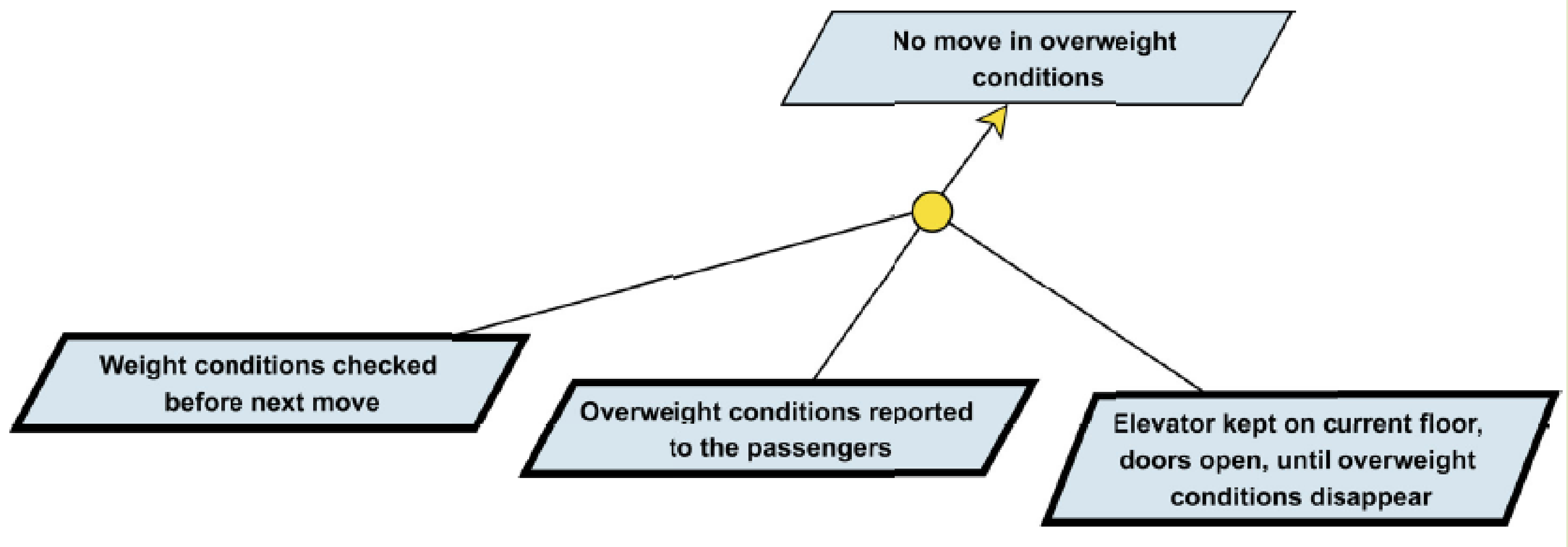


Generic NFR *Goal* for a Safe System

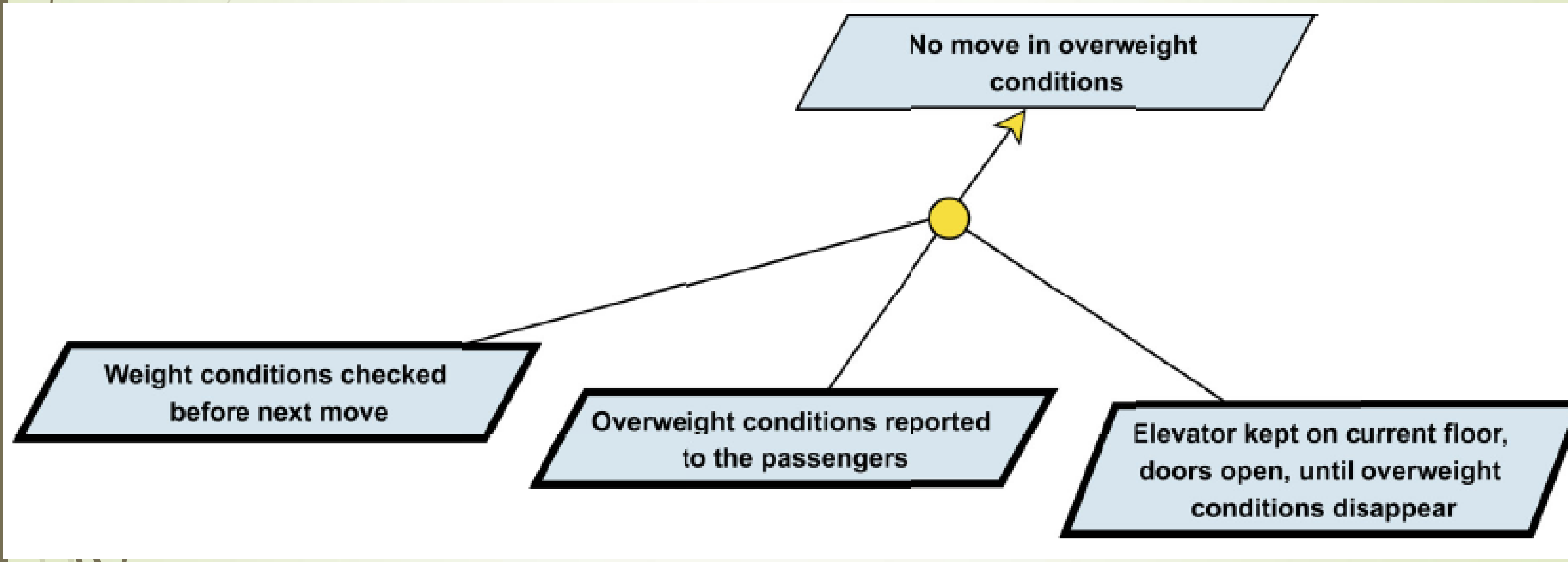



Safe elevator system
No casualties
Robust and reliable elevator system
No intrusion
Control room lockable
Software secured
Control room locked
Safe stay inside the cage
No door opening while moving
No passenger locked in forever
contact between
ers and moving parts
Lights always on
No move in overweight conditions

Safe elevator system



Safe elevator system: Refinement of NFR: No move in overweight conditions



Concrete model for NFR Safety: Elevator System protected against power failure

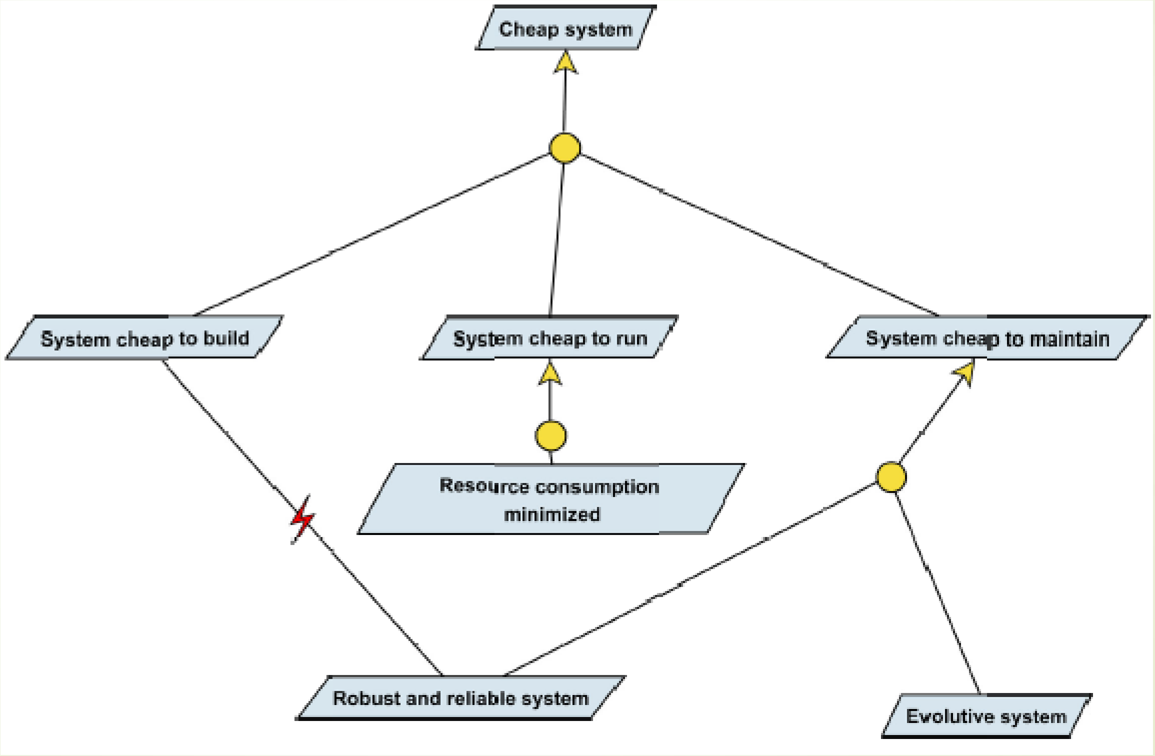
Conflicting goals

- When it is not possible to completely satisfy two goals simultaneously
  - Performance goals may conflict with safety goals
  - Information goals may conflict with security and privacy goals
- Dealing with conflicts (or more generally, with obstacles) allows
  - to build a more complete requirements document and
  - To build a more robust system
- Obstacles prevent goals from being achieved
  - Defensive approach

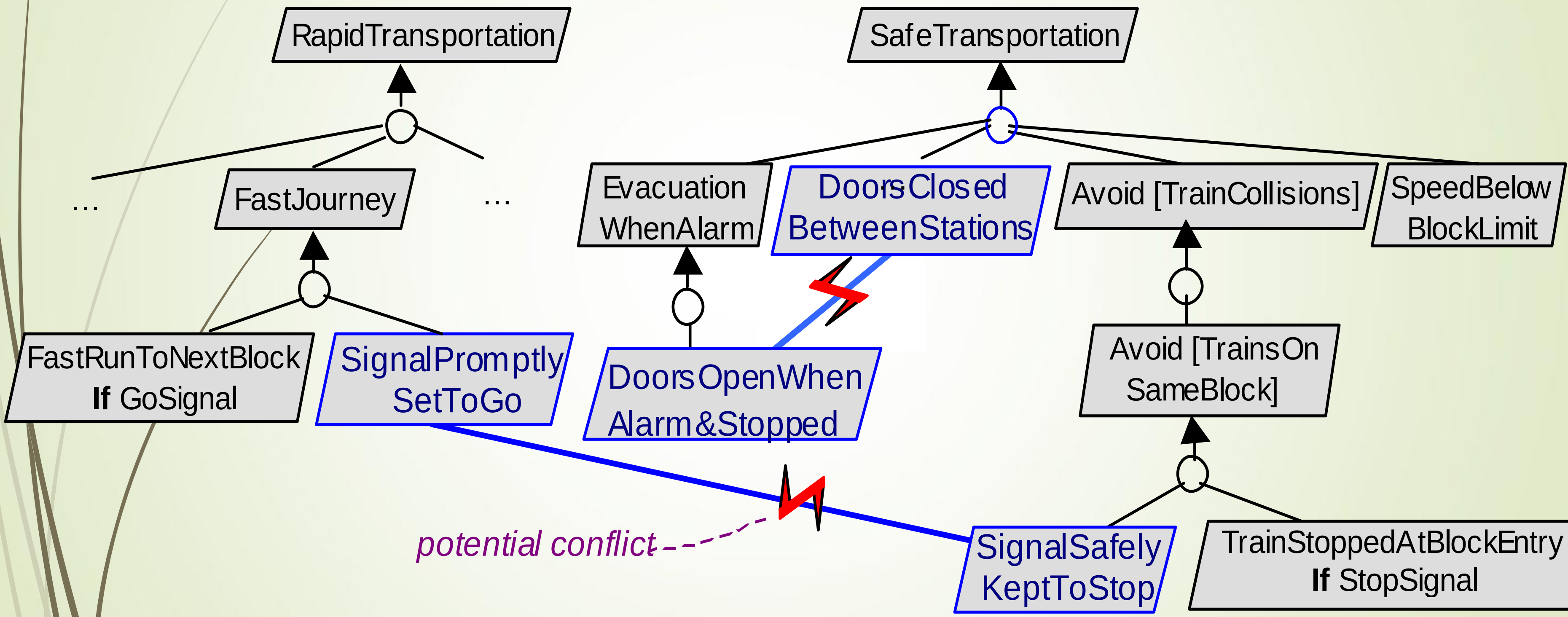
Conflict management

- When conflicts are detected:
  - Negotiation to conflict resolution
    - Select alternatives or
    - Re-evaluate the priorities or
    - Revise requirements

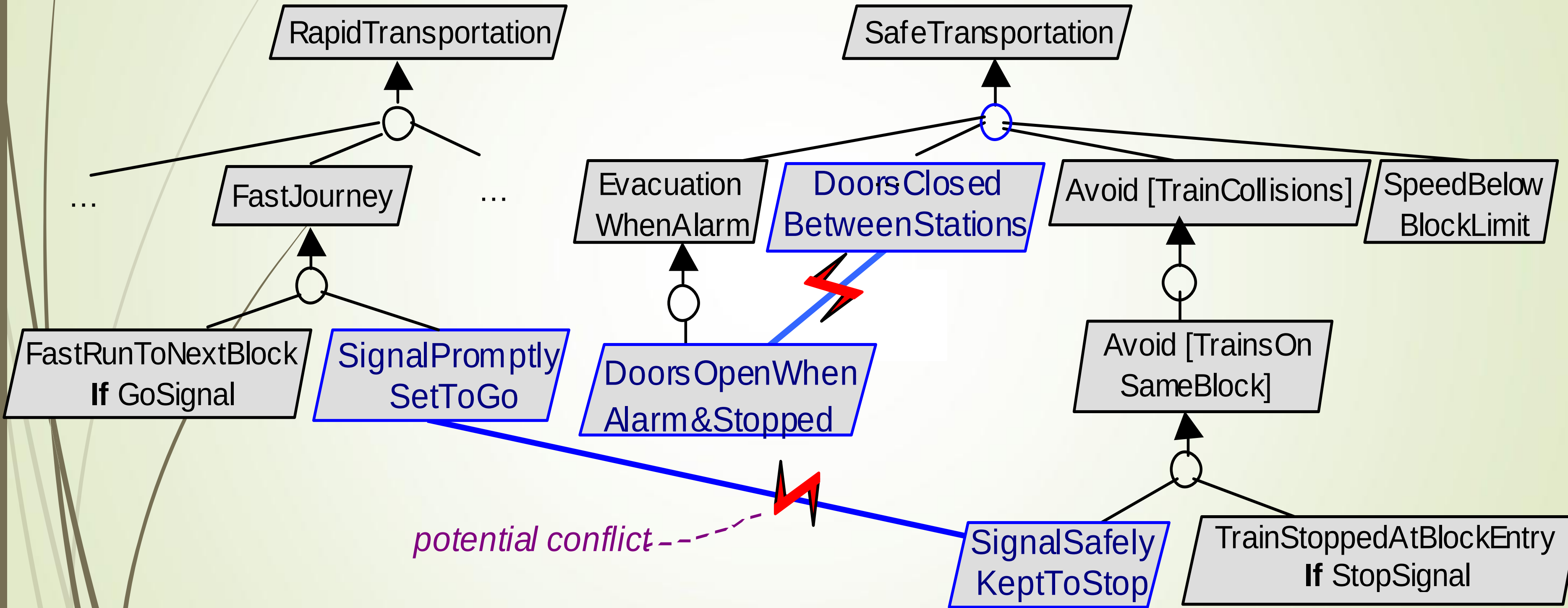
Conflict Identification: Generic goal for economic aspects of a system

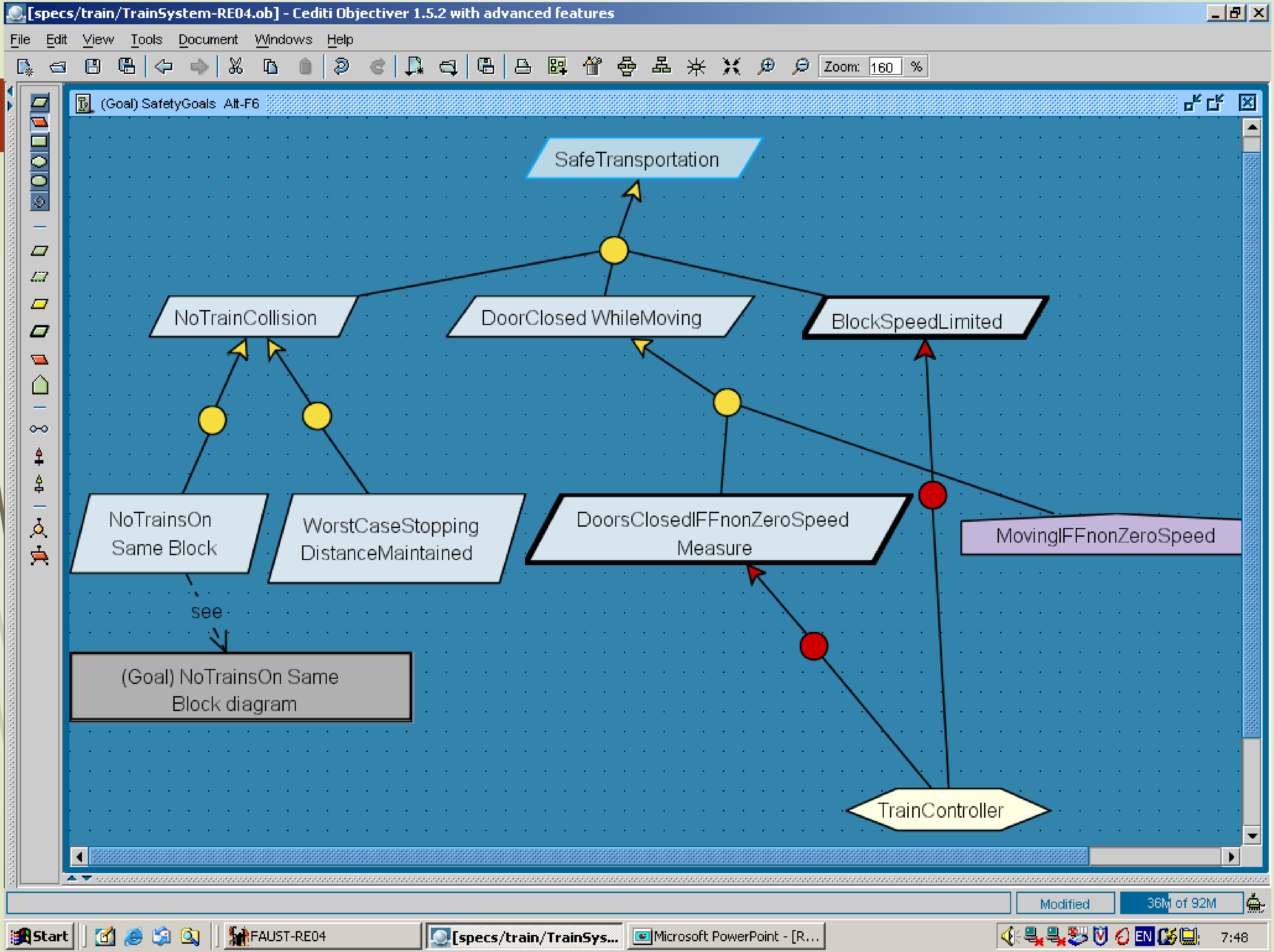


Capturing potential conflicts among goals

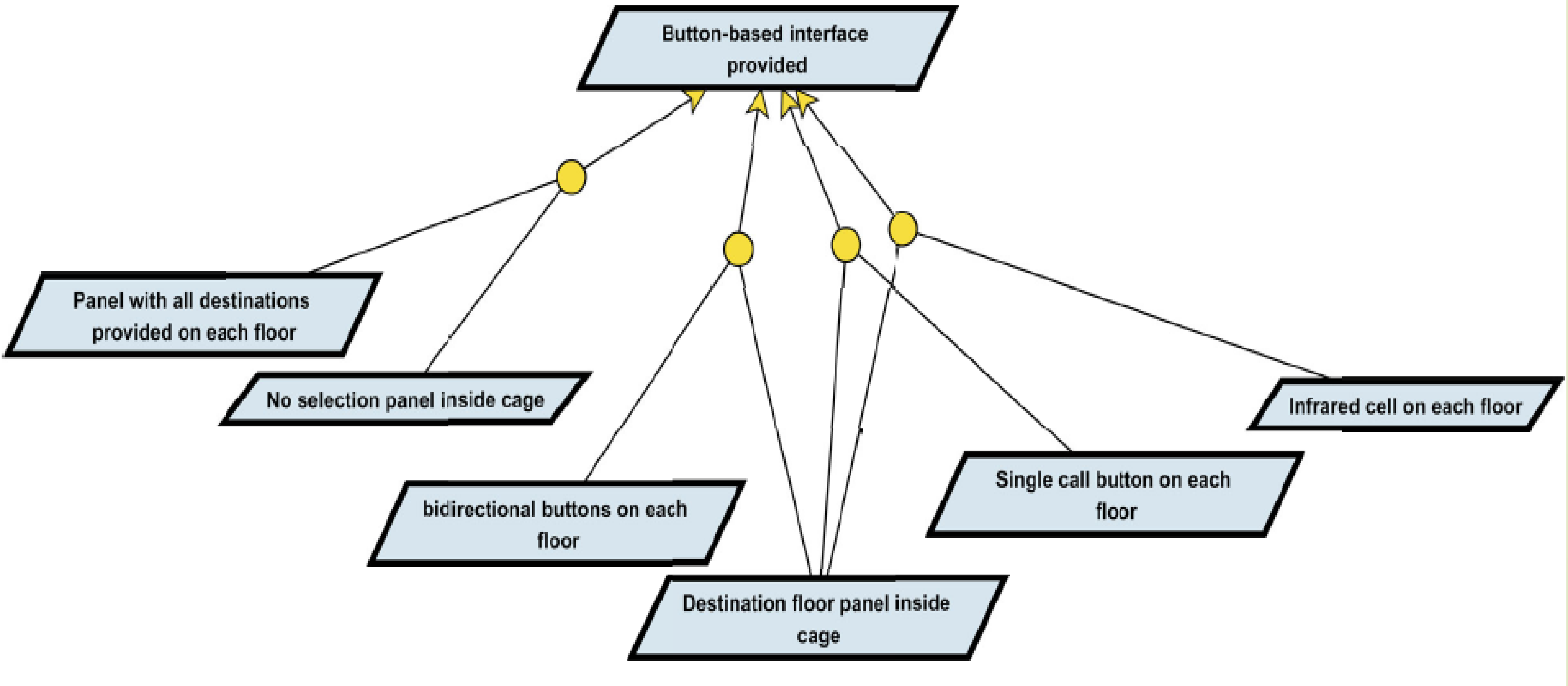


Capturing potential conflicts among goals

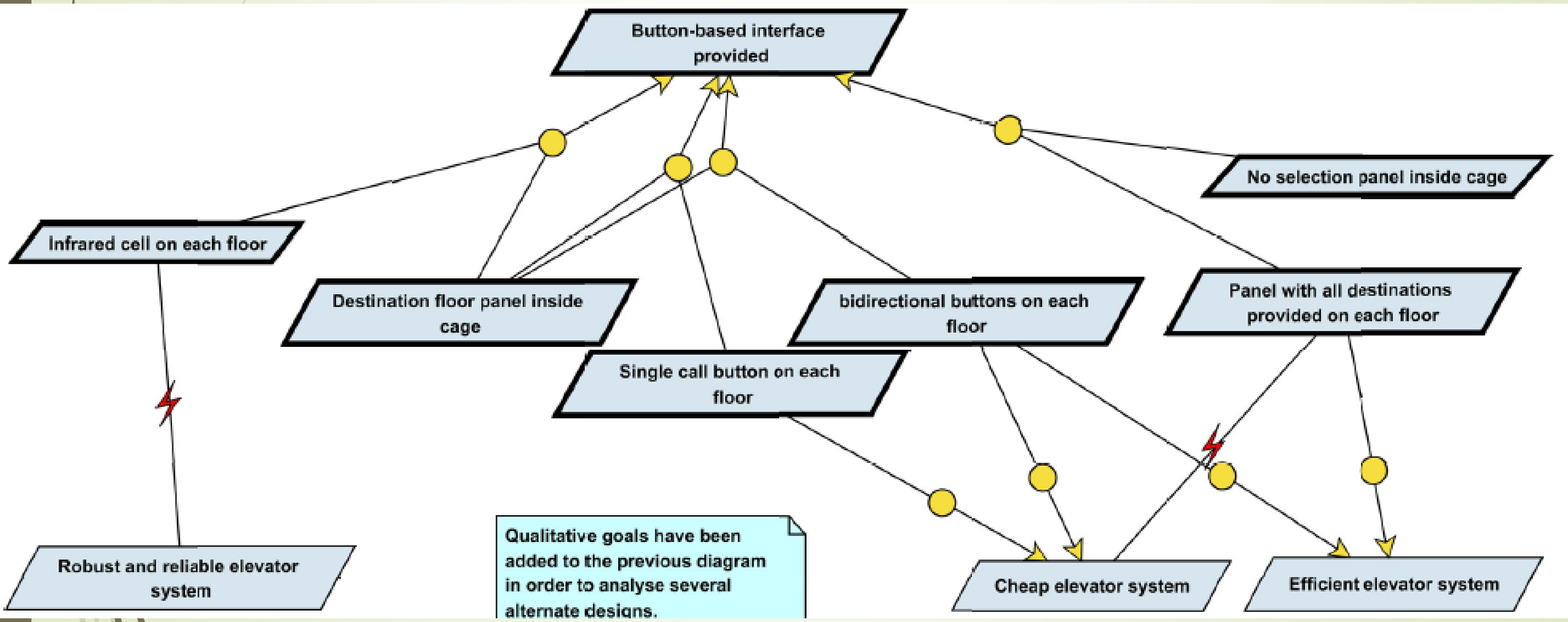



[specs/train/TrainSystem-RE04.ob] - Cediti Objectiver 1.5.2 with advanced features
File Edit View Tools Document Windows Help
Zoom: 160 %
(Goal) SafetyGoals Alt-F6
SafeTransportation
NoTrainCollision
DoorClosed WhileMoving
BlockSpeedLimited
NoTrainsOn Same Block
WorstCaseStopping DistanceMaintained
DoorsClosedIFFnonZeroSpeed Measure
MovingIFFnonZeroSpeed
see
(Goal) NoTrainsOn Same Block diagram
TrainController
Modified
36M of 92M
Start
FAUST-RE04
[specs/train/TrainSys...
Microsoft PowerPoint - [R...
EN
7:48

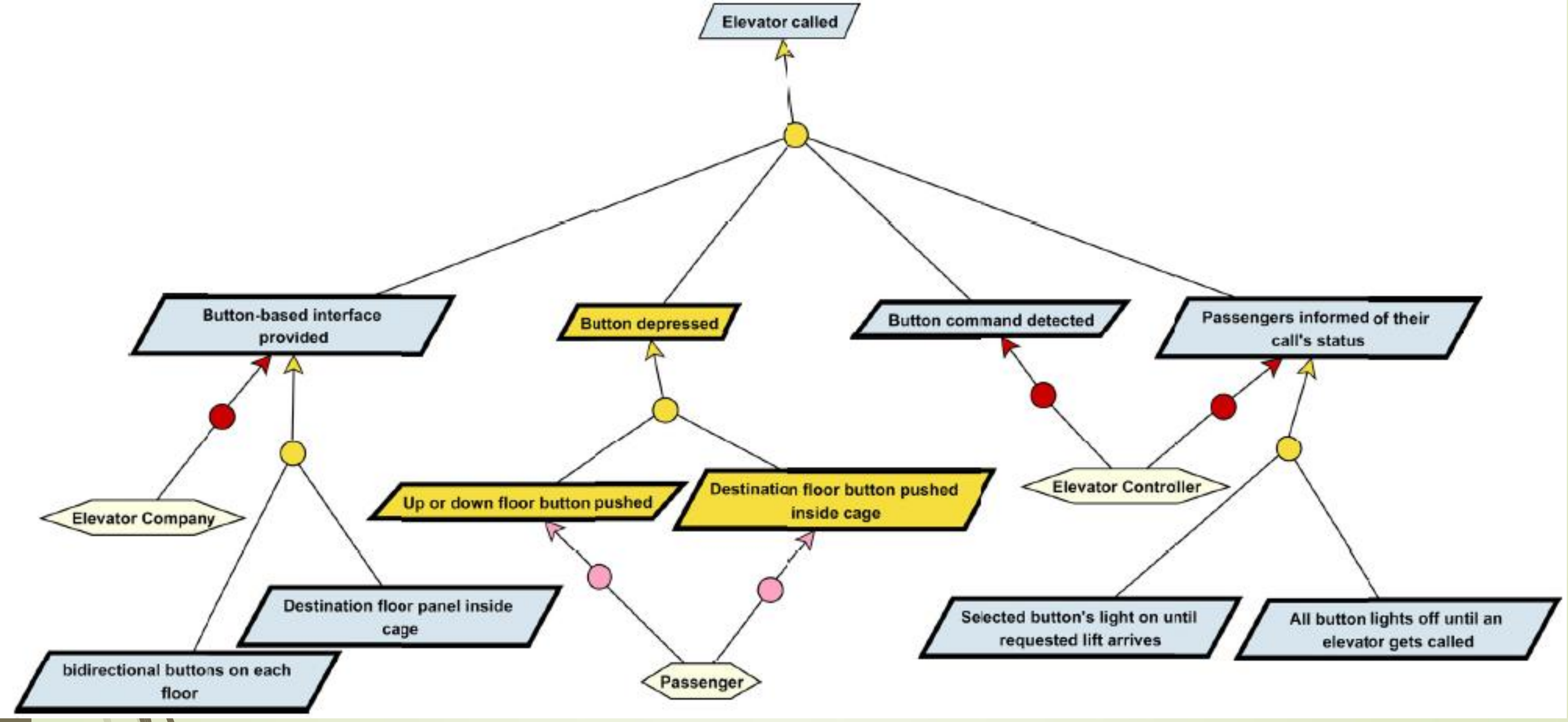
Alternatives



Analysis of the button based interface with NF goals



Elevator called with selected alternatives

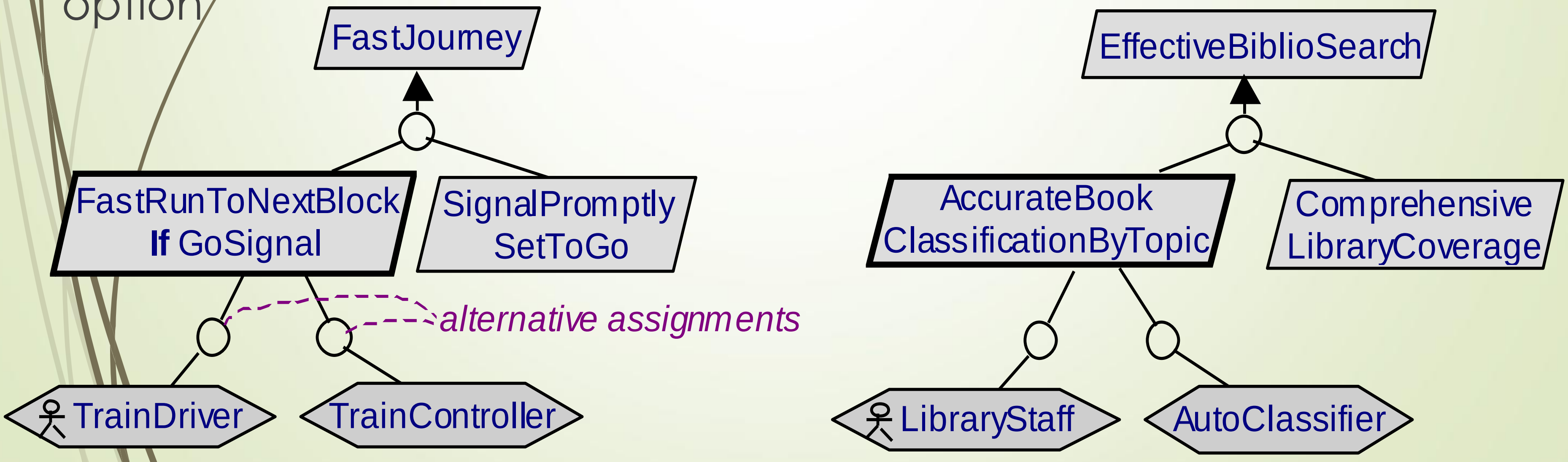


Capturing options: alternative assignments

- An **OR-assignment** of goal G to agents $A_1, ..., A_m$ states that G can be satisfied by behavioral restrictions of any of the alternative agents A_i
- Alternative assignments yield different system proposals

(e.g. different degrees of automation)

- Pros/cons to be evaluated against soft goals for selection of best option



What are obstacles ?

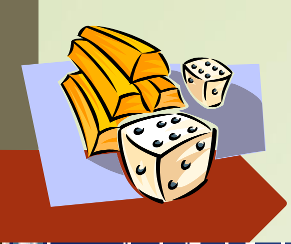
- Motivation: goals in refinement graph are often too ideal, likely to be violated under abnormal conditions (unintentional *or* intentional agent behaviors)
- **Obstacle** = condition on system for violation of corresponding assertion (generally a goal)
  - {O, *Dom* } |= not *G* *obstruction*

e.g. *G:* TrainStoppedAtBlockSignal **If** StopSignal

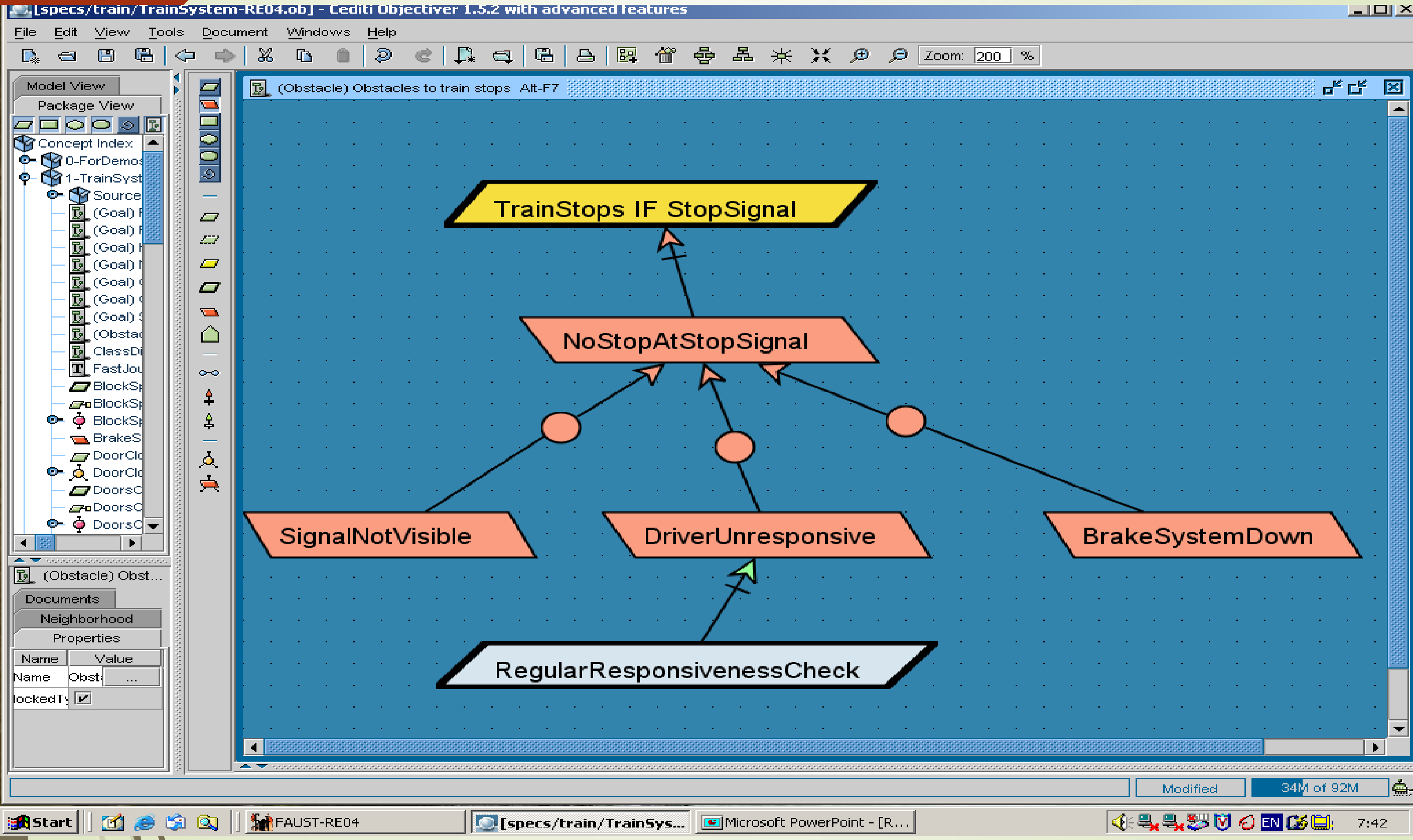
Dom: **If** TrainStopsAtStopSignal **then** DriverResponsive

O: Driver**Un**responsive

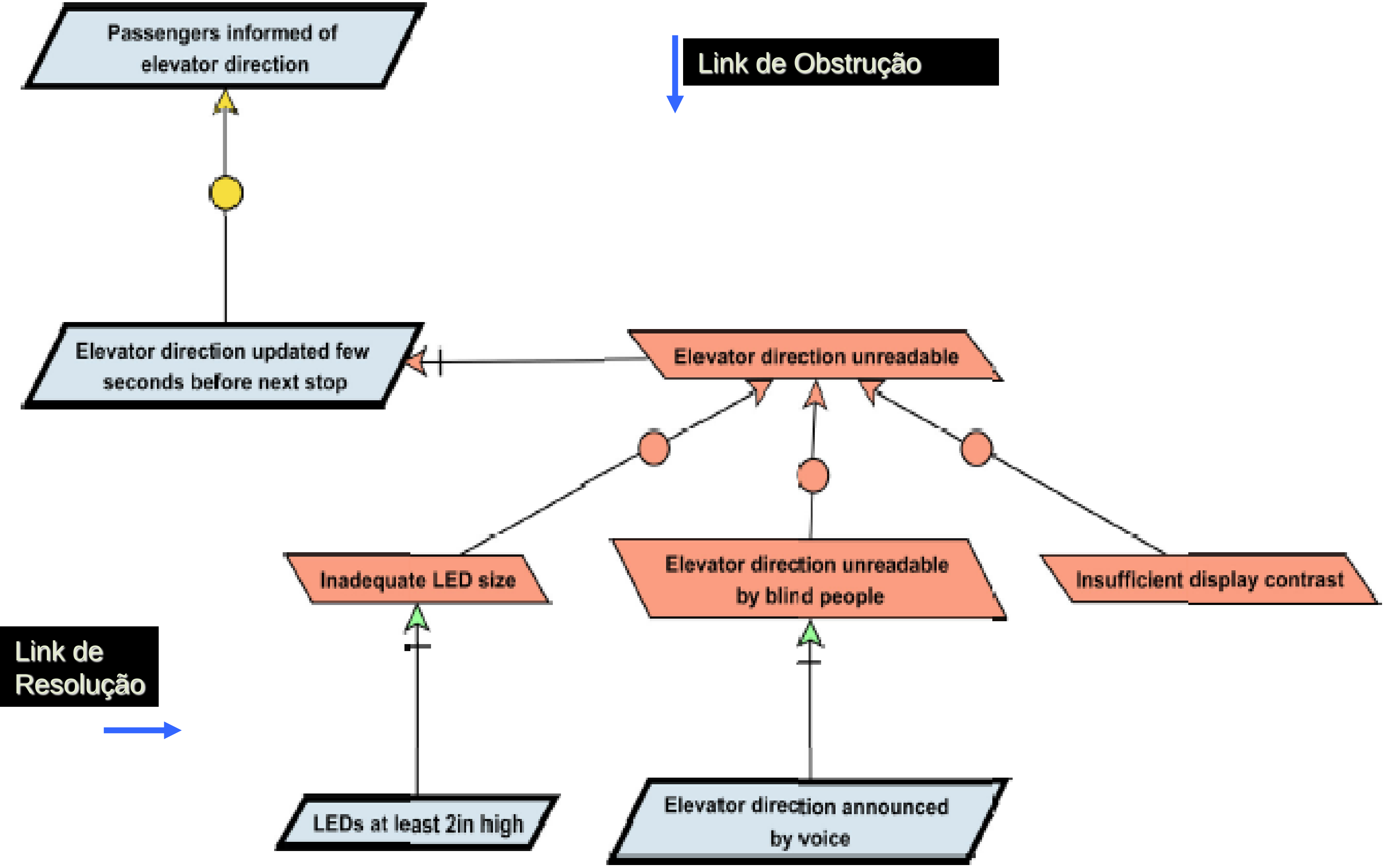
- For behavioral goal: existential property capturing unadmissible behavior (**negative** scenario)



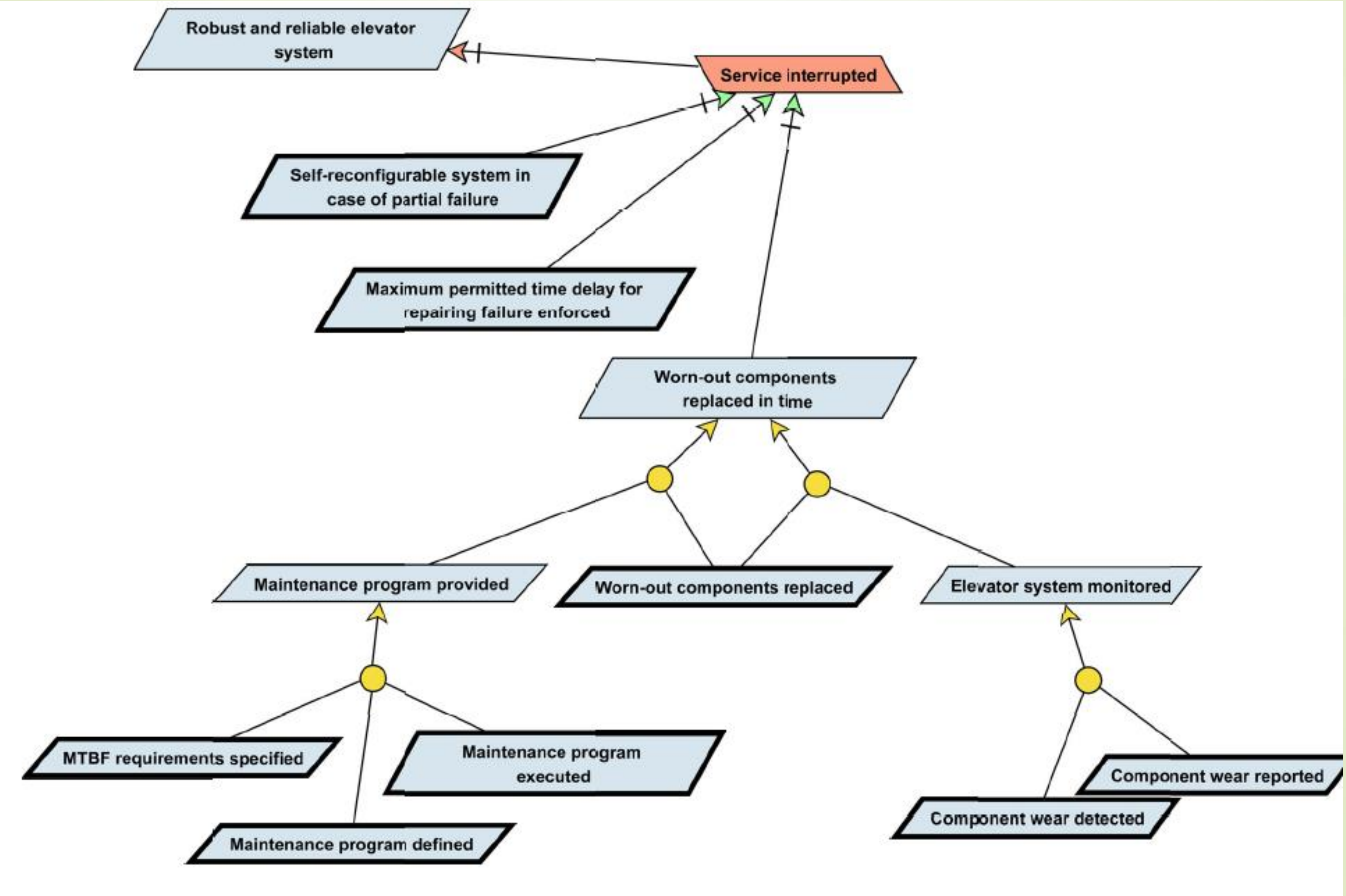
Risk analysis can be anchored on goal models



Obstacles


Passengers informed of elevator direction
Link de Obstrução
Elevator direction updated few seconds before next stop
Elevator direction unreadable
Inadequate LED size
Elevator direction unreadable by blind people
Insufficient display contrast
Link de Resolução
LEDs at least 2in high
Elevator direction announced by voice

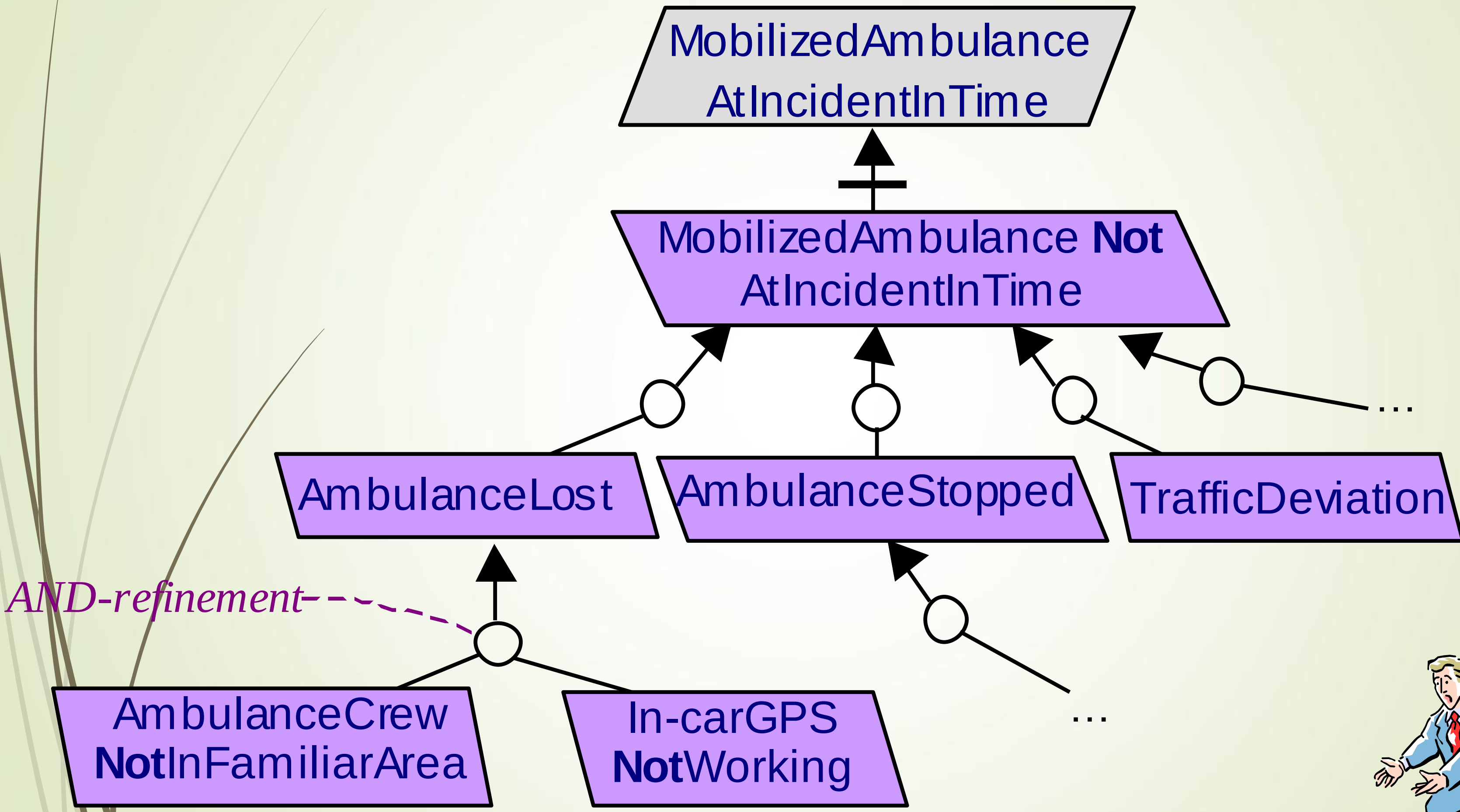
Robust and reliable elevator system



Exercise

- Mobilise Ambulance at incident in time
- Obstacle: Mobilise Ambulance NOT at incident in time

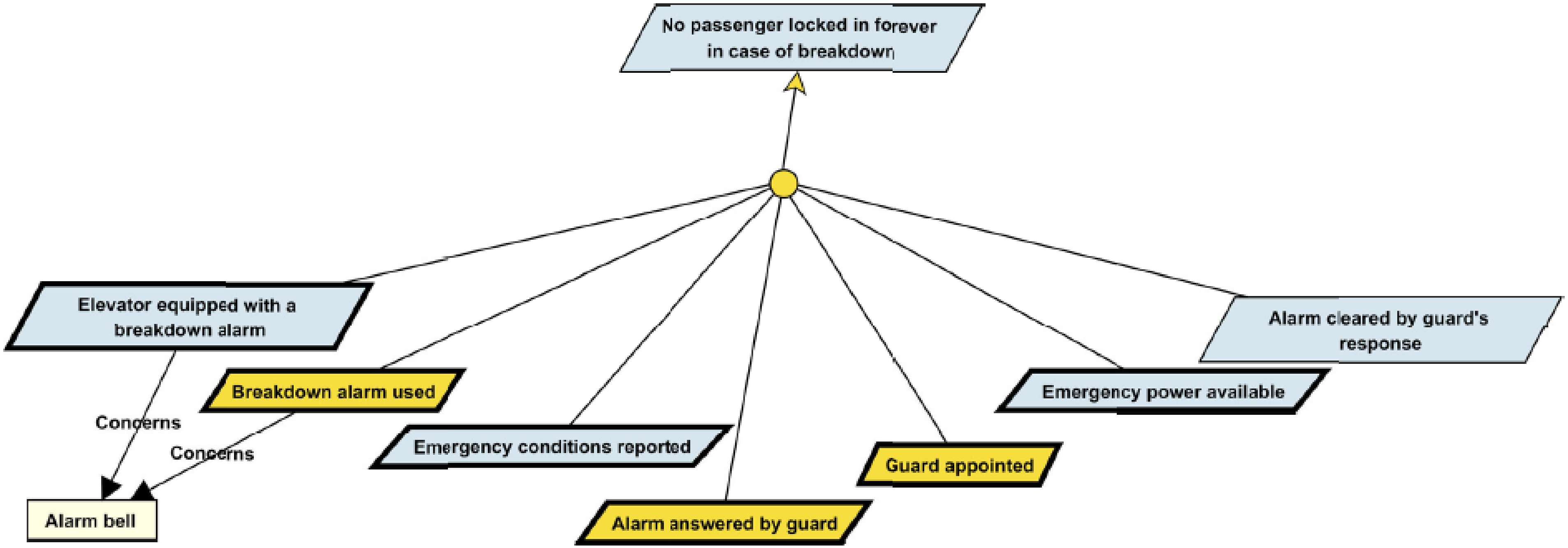
Obstacle diagrams as AND/OR refinement trees (2)



Object model

- ***entities***: they represent **independent**, **passive** objects. For instance, *elevator doors*, *buttons*, etc... 'Independent' means that their descriptions needn't refer to other objects of the model. They may have attributes whose values define a set of states the entity can transition to. They are 'passive' means they can't perform operations.
- ***agents***: represent **independent**, **active** objects. For instance, *elevator company, passenger,* and *elevator controller* are agents. They are active meaning they can perform operations. Operations usually imply state transitions on entities (for instance, the "CloseDoor" operation implies the following state transition on the entity "Door": status attribute changed from "Open" to "Close").
- ***associations***: are **dependent**, **passive** objects. 'Dependent' because their descriptions refer to other objects. For instance, the *At* association links a *Cage* to a *Floor*.

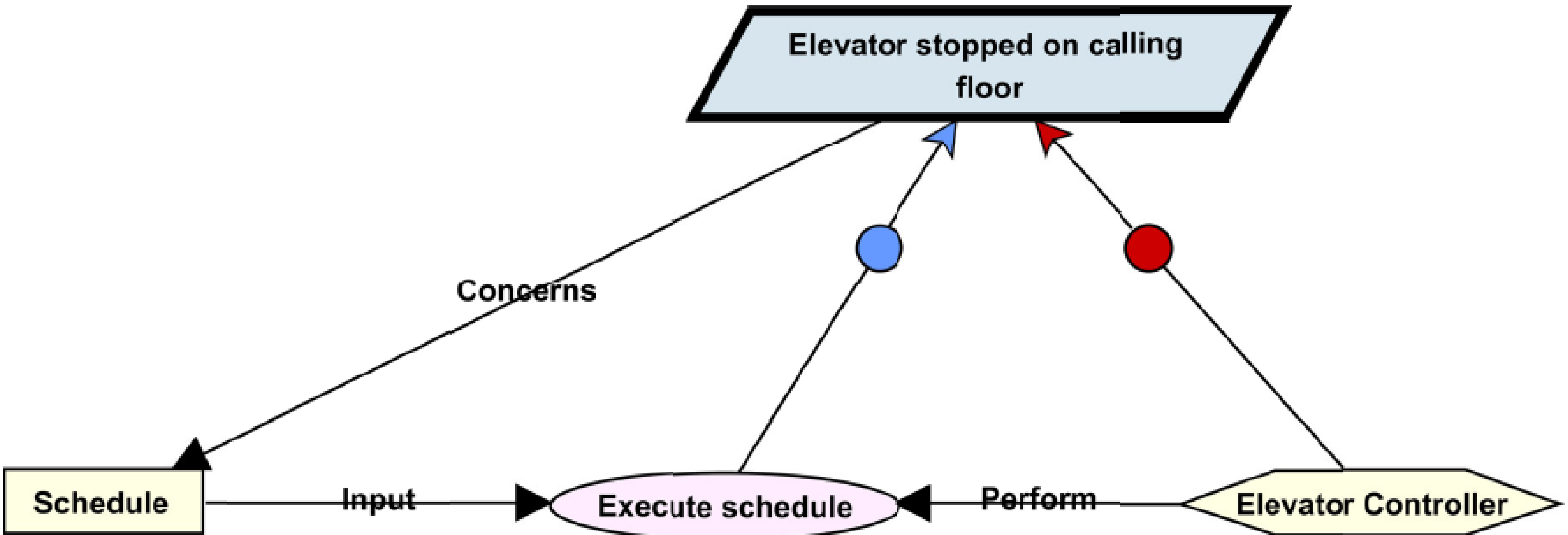
Objects


No passenger locked in forever in case of breakdown
Elevator equipped with a breakdown alarm
Alarm cleared by guard's response
Breakdown alarm used
Emergency power available
Concerns
Concerns
Emergency conditions reported
Guard appointed
Alarm bell
Alarm answered by guard

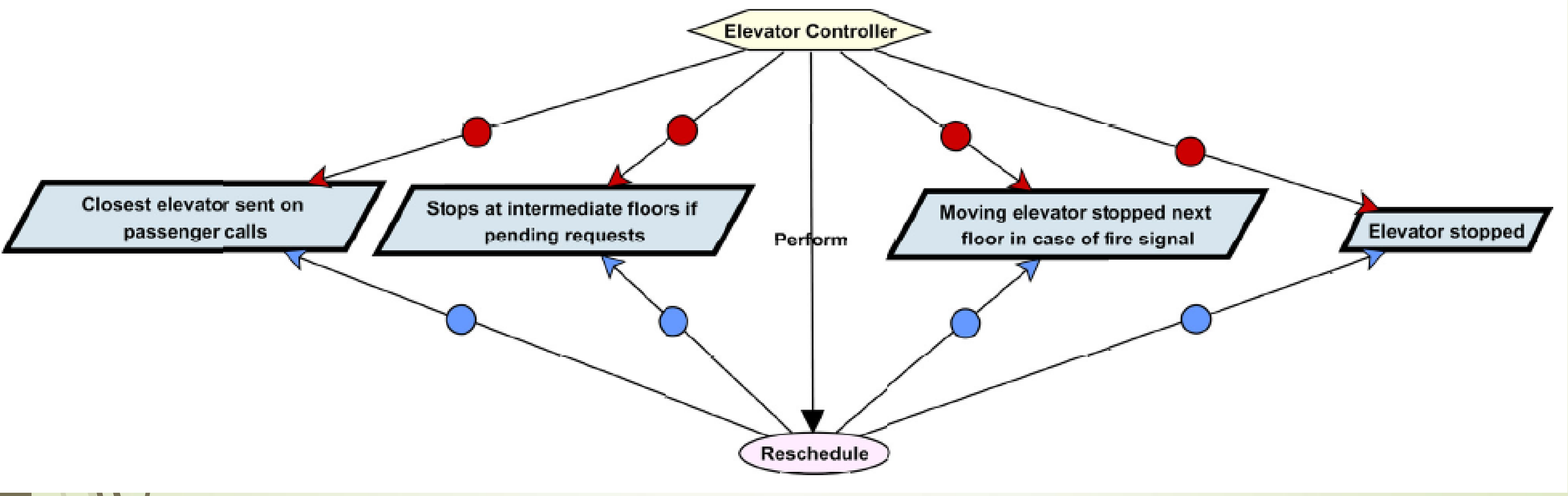
Operation Model (simplified)

- This model describes all teh behaviors that agents need to fulfill their reqs.
- Behaviors are expressed in terms of operations performed by the agents
- Operations work on objects defined in the object model
- Operations are used to operationalize or fulfill reqs

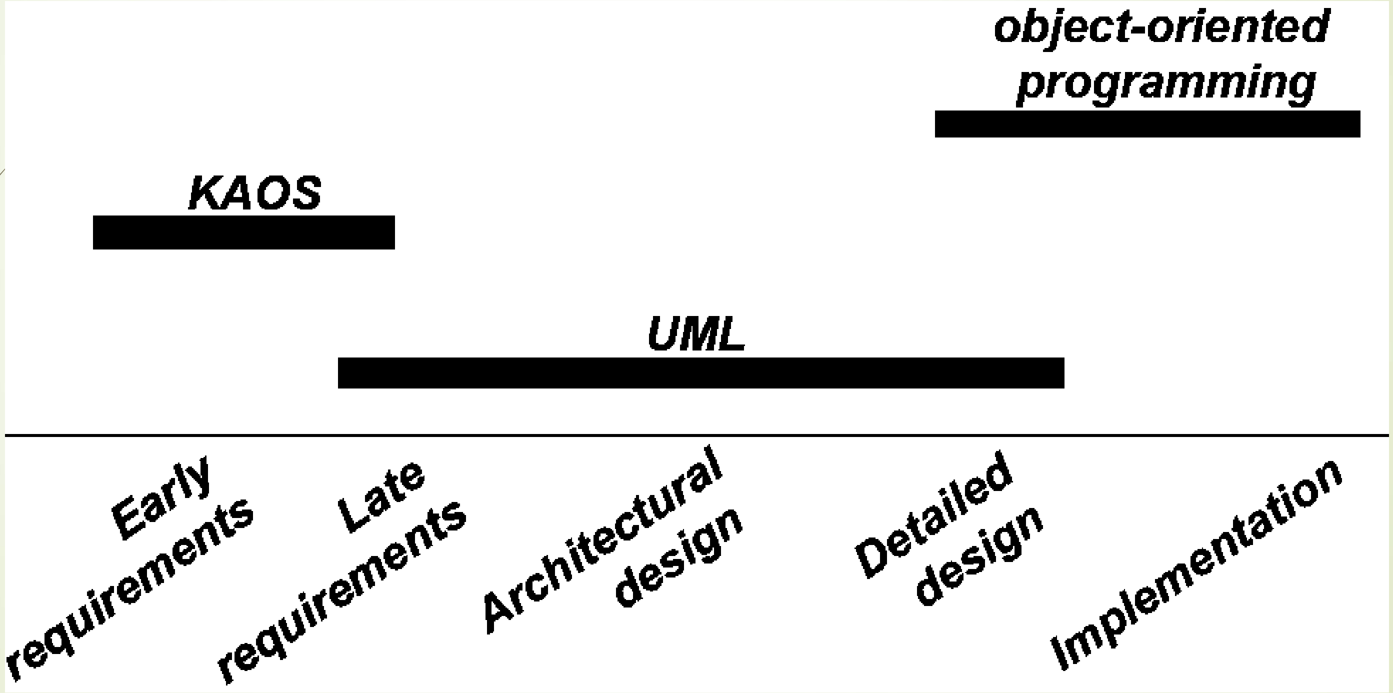
Responsibility-Operationalization-performance



Operationalization Reschedule



Goals and UML



KAOS tools and other goal-oriented approaches

- KAOS tools
  - Grail
  - Objectiver
- i* (istar)
- Tropos
- GRL

Bibliography

- Lamsweerde, Axel van; "*Goal-Oriented Requirements Engineering: A Guided Tour*"; *Université Catholique de Louvain; Louvain-la-Neuve; Belgium; 2001*
- "A KAOS Tutorial"; Objectiver; 2003
- Lamsweerde, Axel van; "*Goal-Oriented Requirements Engineering: From System Objectives to UML Models to Precise Software Specification*"; *Université Catholique de Louvain; Louvain-la-Neuve; Belgium; May 2003*
- Lamsweerde, Axel van; "*Requirements Engineering - - from system goals to UML models to software specifications*"; *Wiley, 2009*

Exercício

- SmartHome
  - Para satisfazer a area de consumo de energia é preciso configurar e minimizar o desperdício.
  - Pode-se configurar ar-condicionado e luzes
  - Contra o desperdício pode-se controlar luzes e termostato e controlar máquinas como máquina de cozinha.