LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Übung 4 – KAOS

Methoden des Software Engineering
WS 2012/13

Christian Kroiß

Outline

I. More details on KAOS

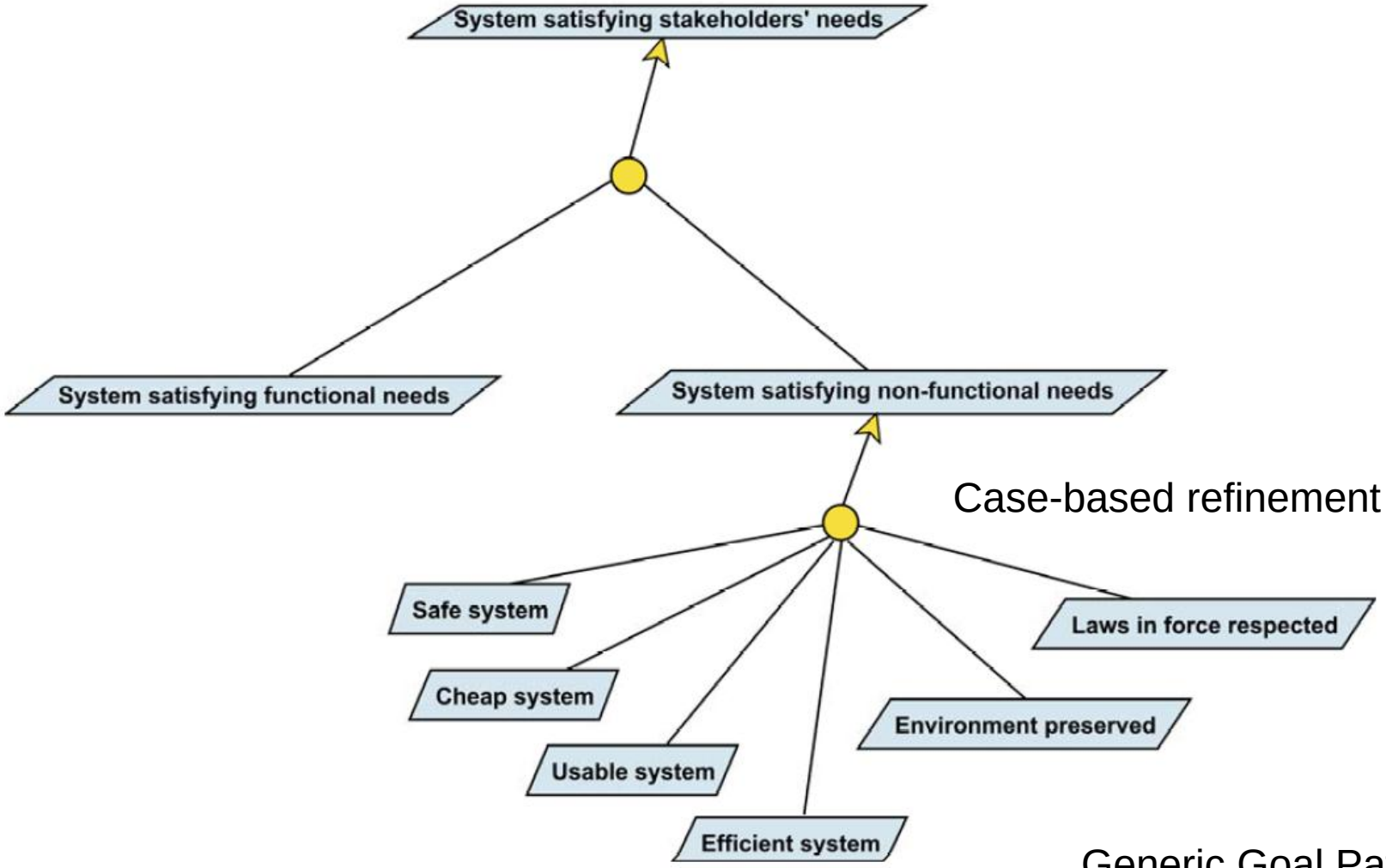
II. Demo of KAOS tool Objectiver

Requirements vs. Software Specifications

- Software-to-be ↔ environment
- 4-variable model

→ cf. [3], slides 13 – 15

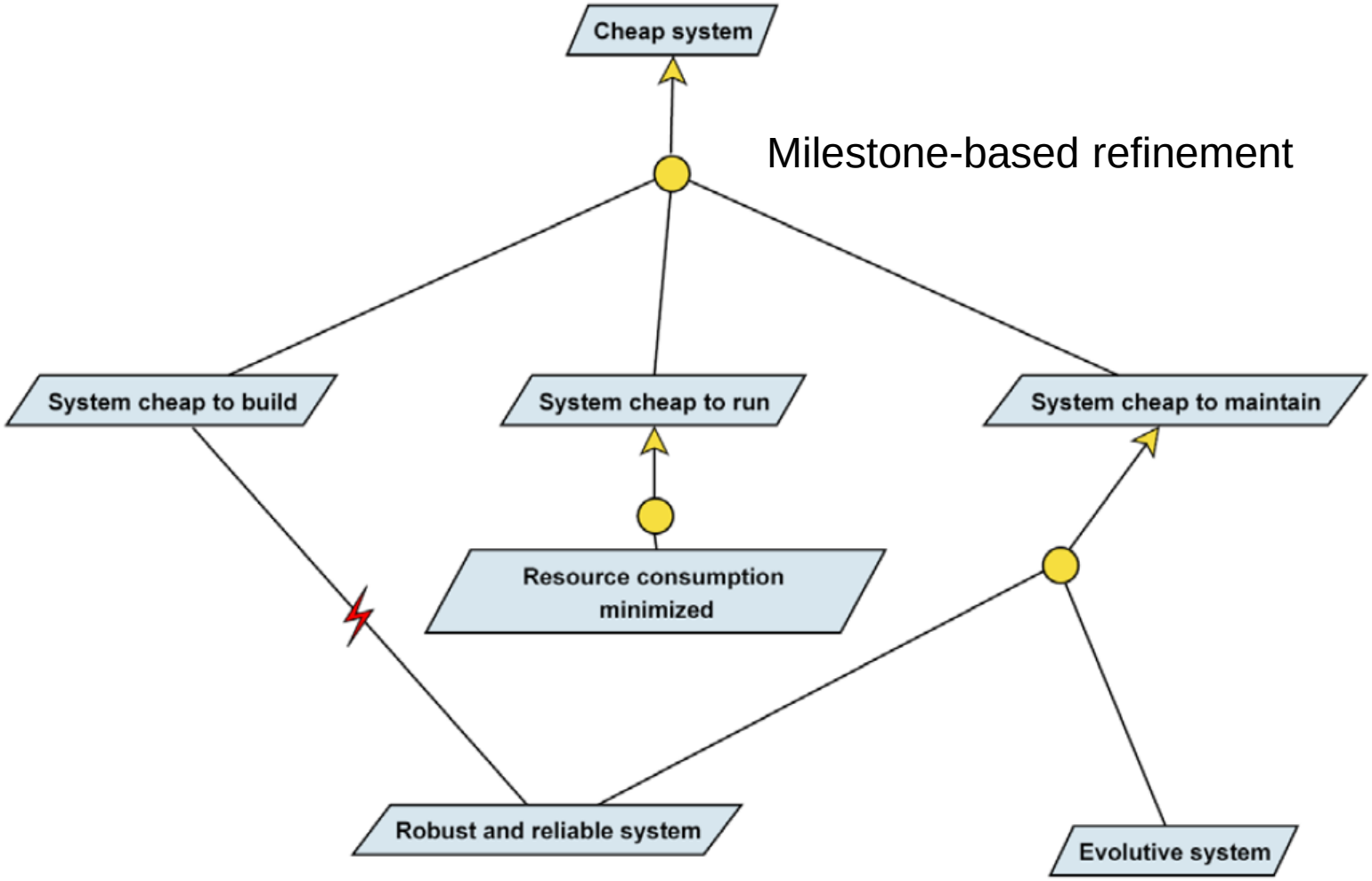
Requirements patterns



Case-based refinement

Generic Goal Pattern

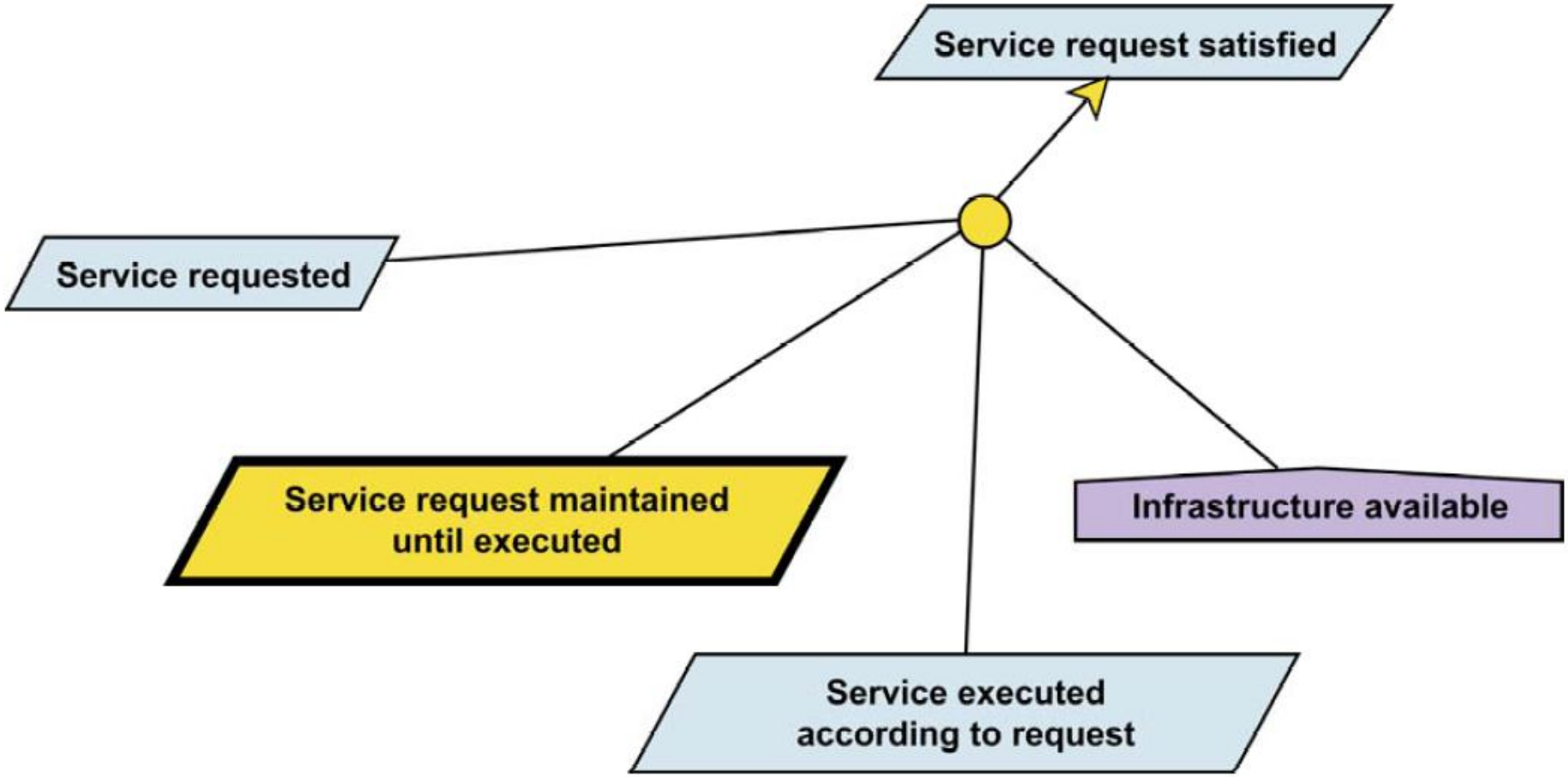
Requirements patterns



Milestone-based refinement

Generic Goal Pattern

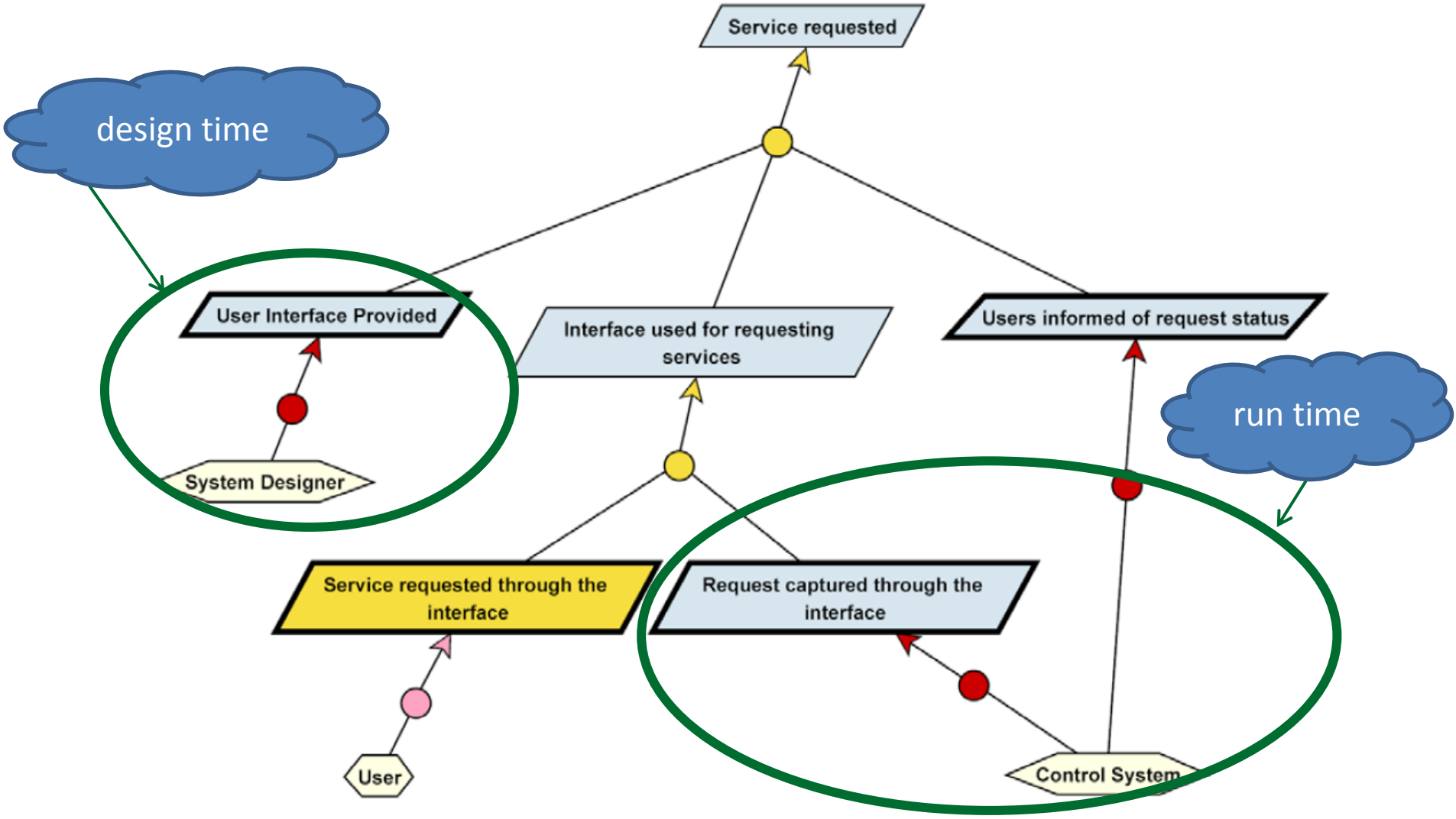
Goals, Expectations, Domain Properties



Goals, Expectations, Domain Properties

- Expectations do not need to be refined further
- Domain properties
  - Domain hypotheses:
    "the elevator system has at least one cage to carry passengers"
  - Domain invariants:
    "the light is either on or off"
- Add qualitative goals
  - Positive contribution: **refinement**
  - Negative contribution: **conflict**

Goals and Requirements

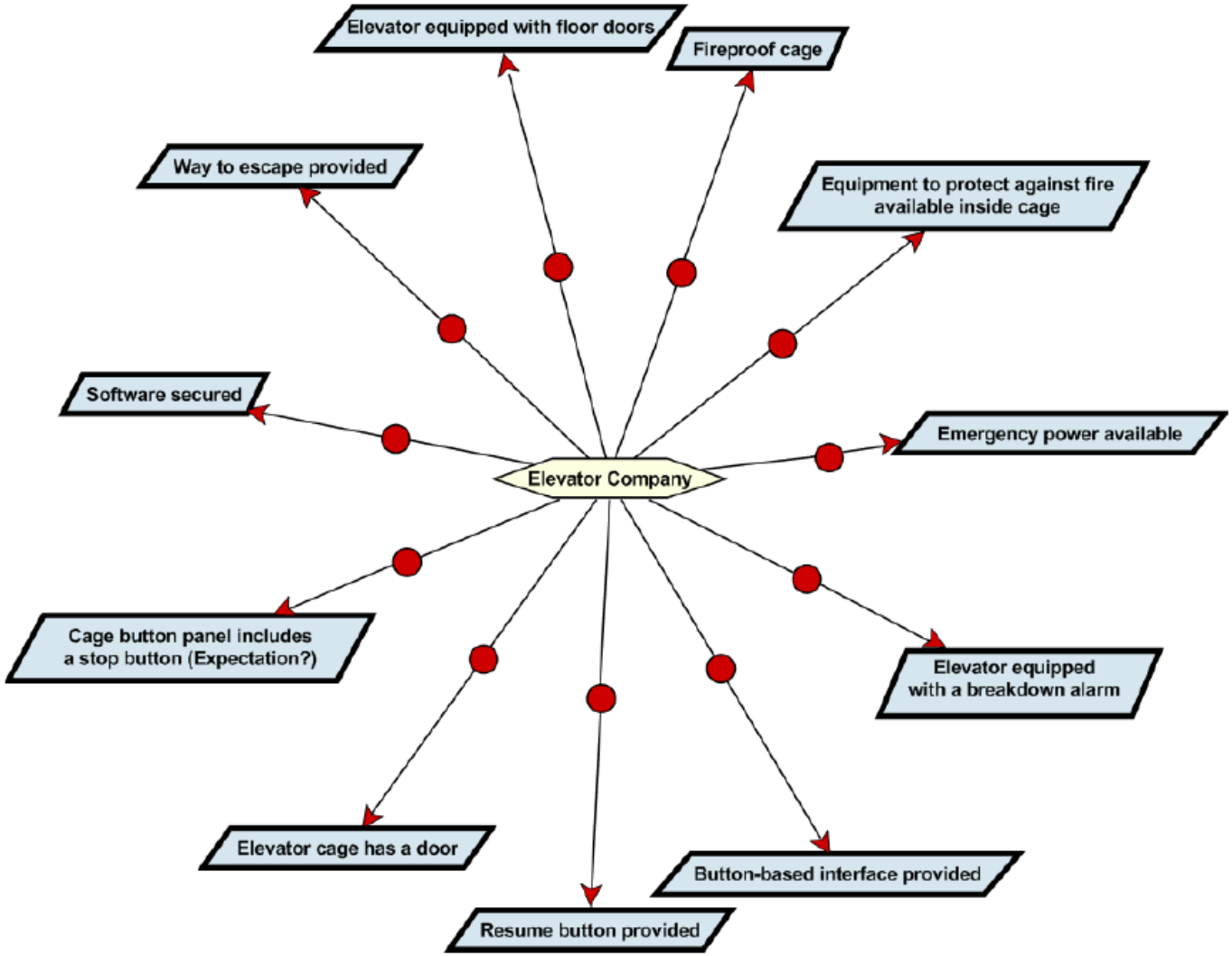


- Goal vs. Requirement
  - A **Requirement** is a goal with associated responsibility
  - Subrequirements are implicitly placed under responsibility of same agent
- Assignment vs. Responsibility
  - **Assignment** is when several agents may be made responsible for some requirement or expectation
  - **Responsibility** is used when there's only one agent who is responsible for it

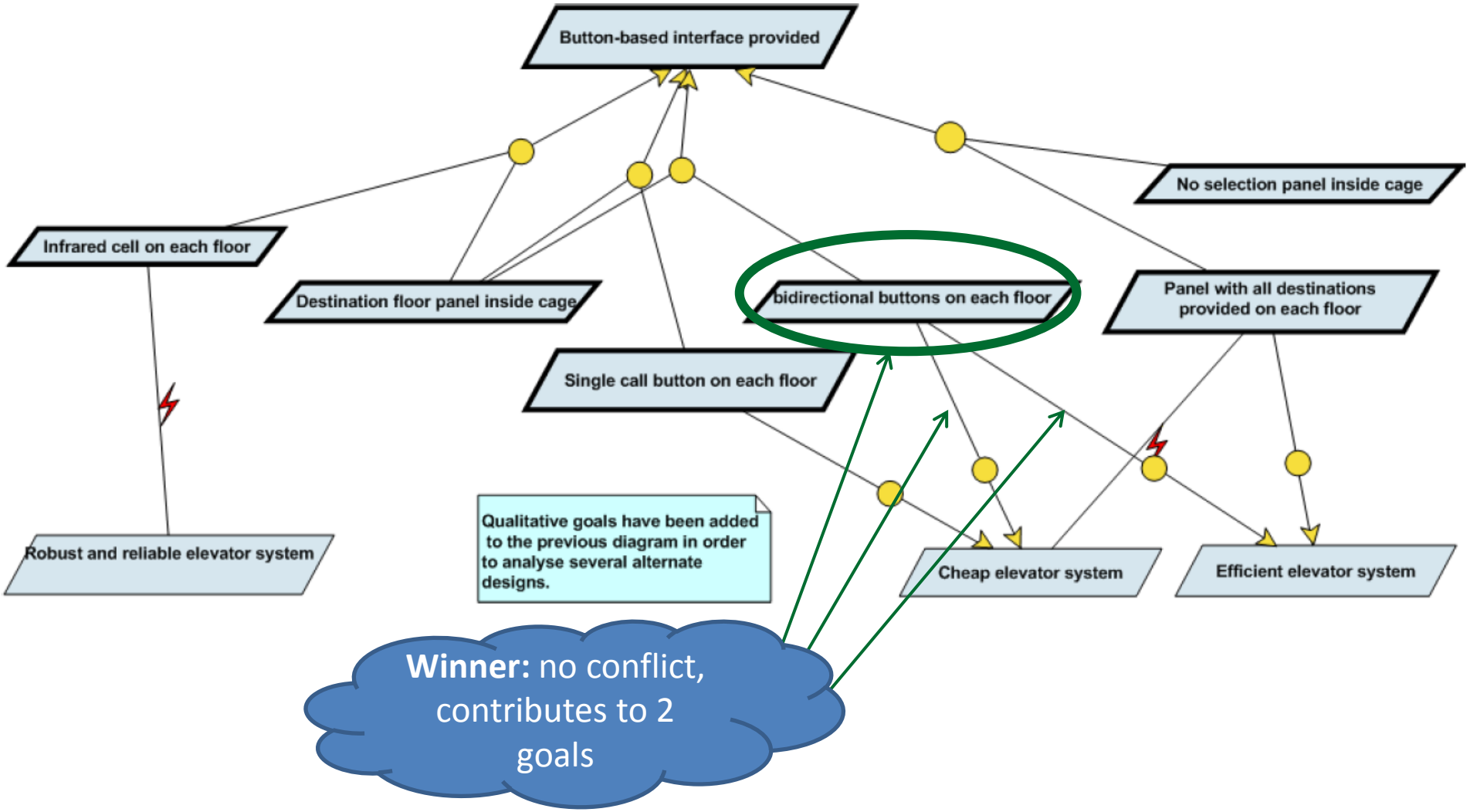
Completeness of Goal Models

- ***Completeness criterion 1:*** *A goal model is said to be complete with respect to the refinement relationship 'if and only if' every leaf goal is either an expectation, a domain property (DomProp) or a requirement.*
- ***Completeness criterion 2:*** *A goal model is complete with respect to the responsibility relationship 'if and only if' every requirement is placed under the responsibility of one and only one agent (either explicitly or implicitly if the requirement refines another one which has been placed under the responsibility of some agent).*

Responsibility Model



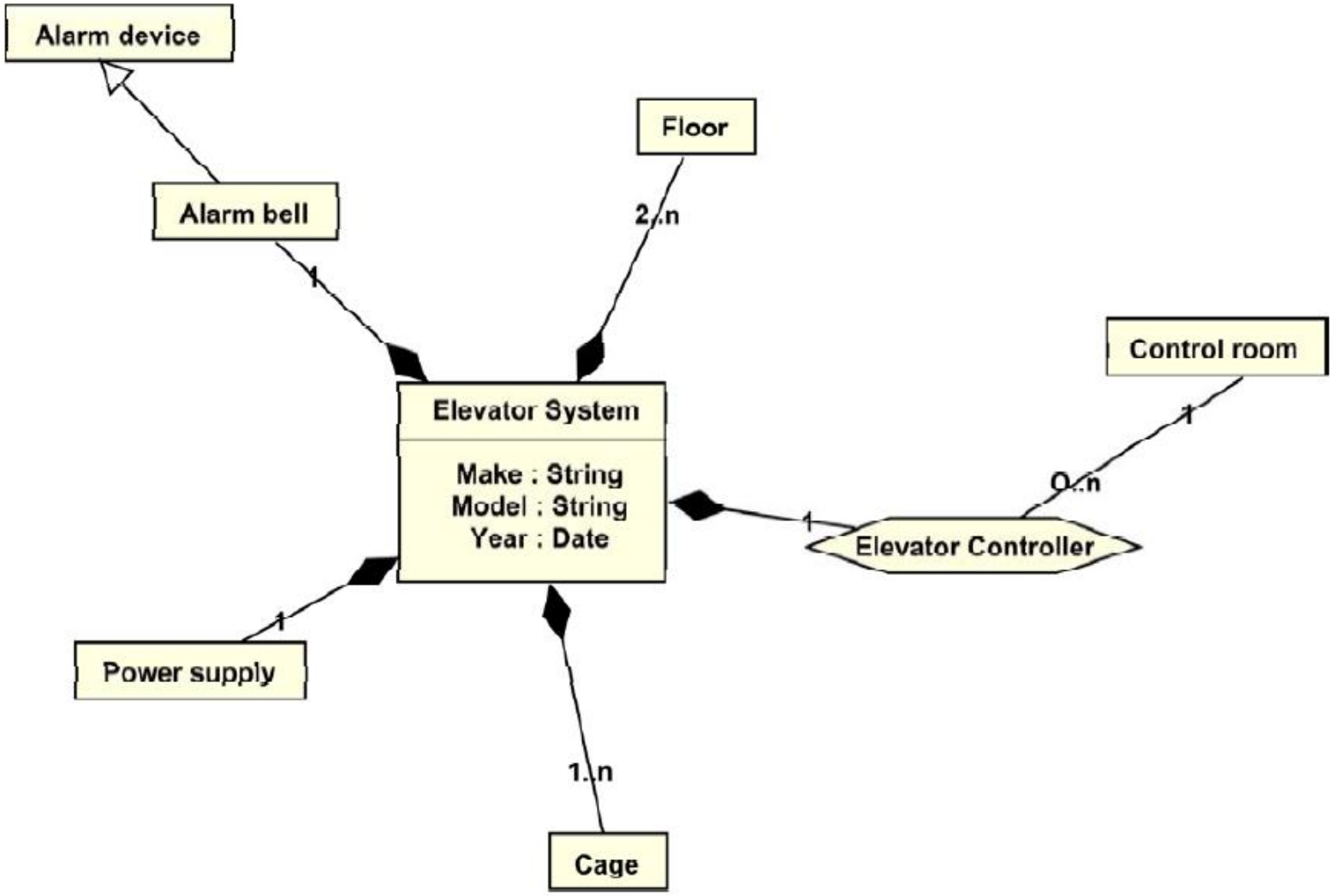
Comparing Alternatives



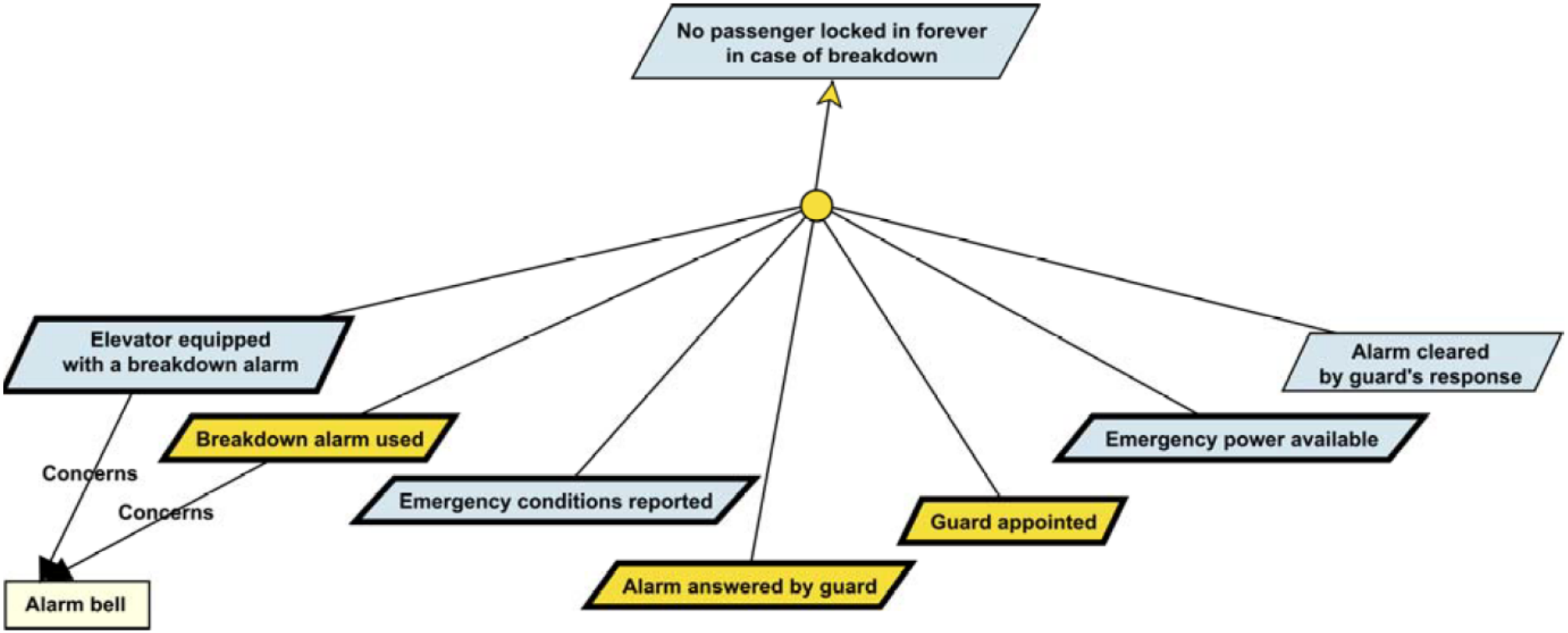
Object Model

- Entities
  - Independent passive objects
  - E.g. Elevator doors, buttons
- Agents
  - Independent & active
  - E.g. Elevator controller, Elevator company
- Associations & Inheritance like in UML

Object Model



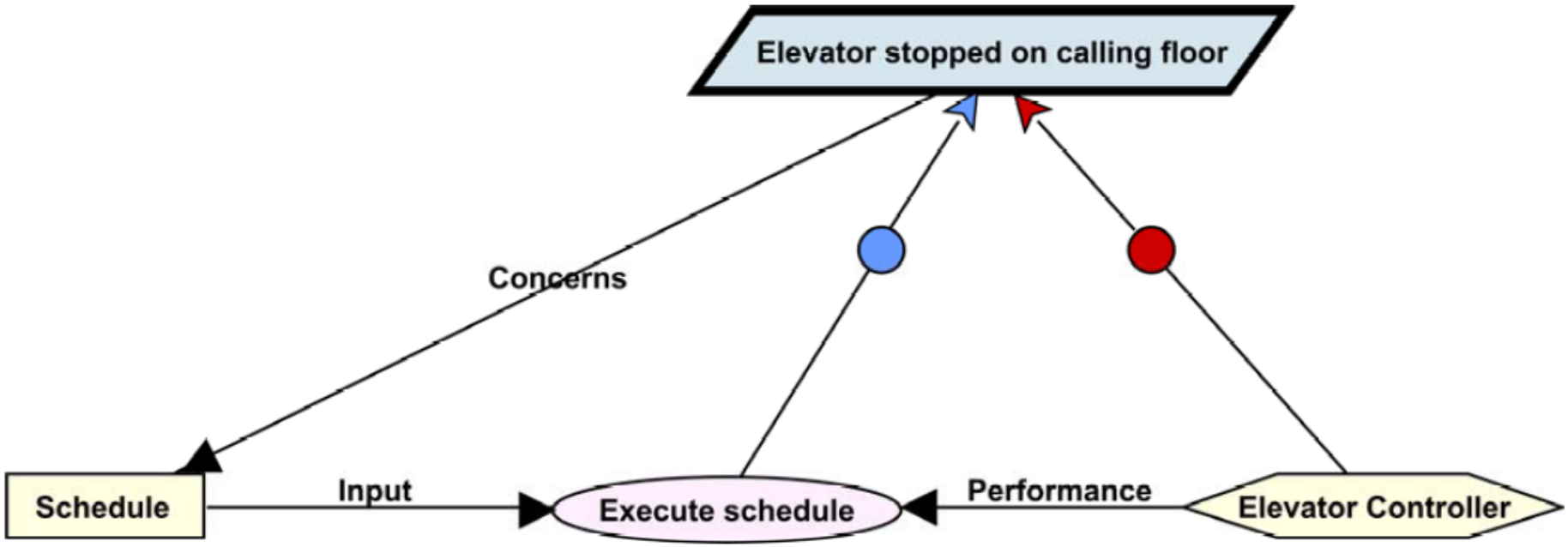
Linking Goals and Objects



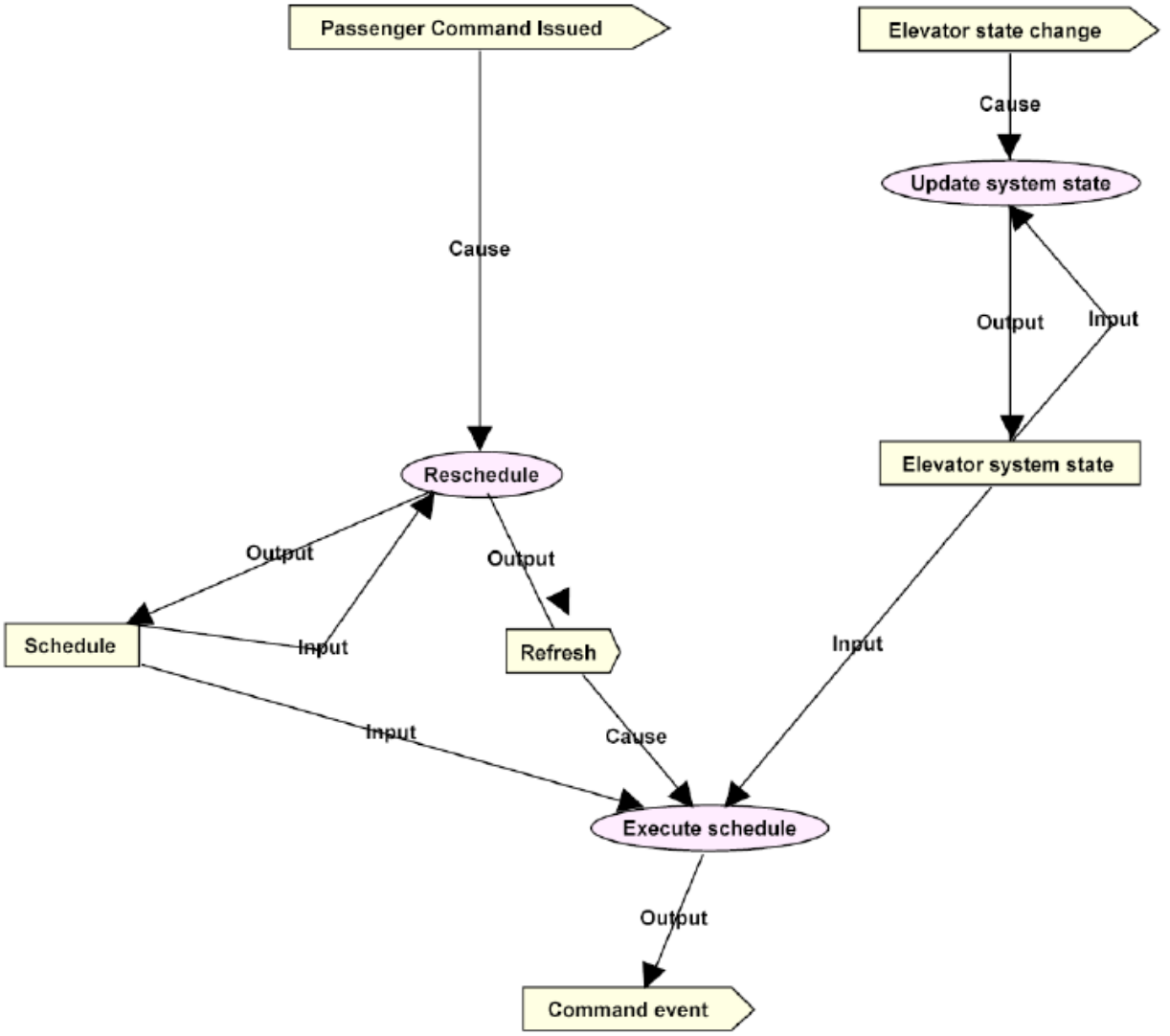
Operation Model

- Requirements are **operationalized** by
  - Objects
  - Operations
- Operations can have **input** & **output**
- Events
  - **cause** operations
  - Can be external or produced by operations (modelled as output)
  - Can be classified by inheritance

Operationalization



Operation Model



Operationalization

- Requirements that are not operationalized are called **open requirements**
  - solutions providers have freedom to address req. As they want
- Operationalization bridges gap between **problem description space** and **solution description space**

> *"A good requirements document (and the underlying KAOS model) describes the problem as completely as possible, while the specification of a solution is limited only to what's really necessary."*

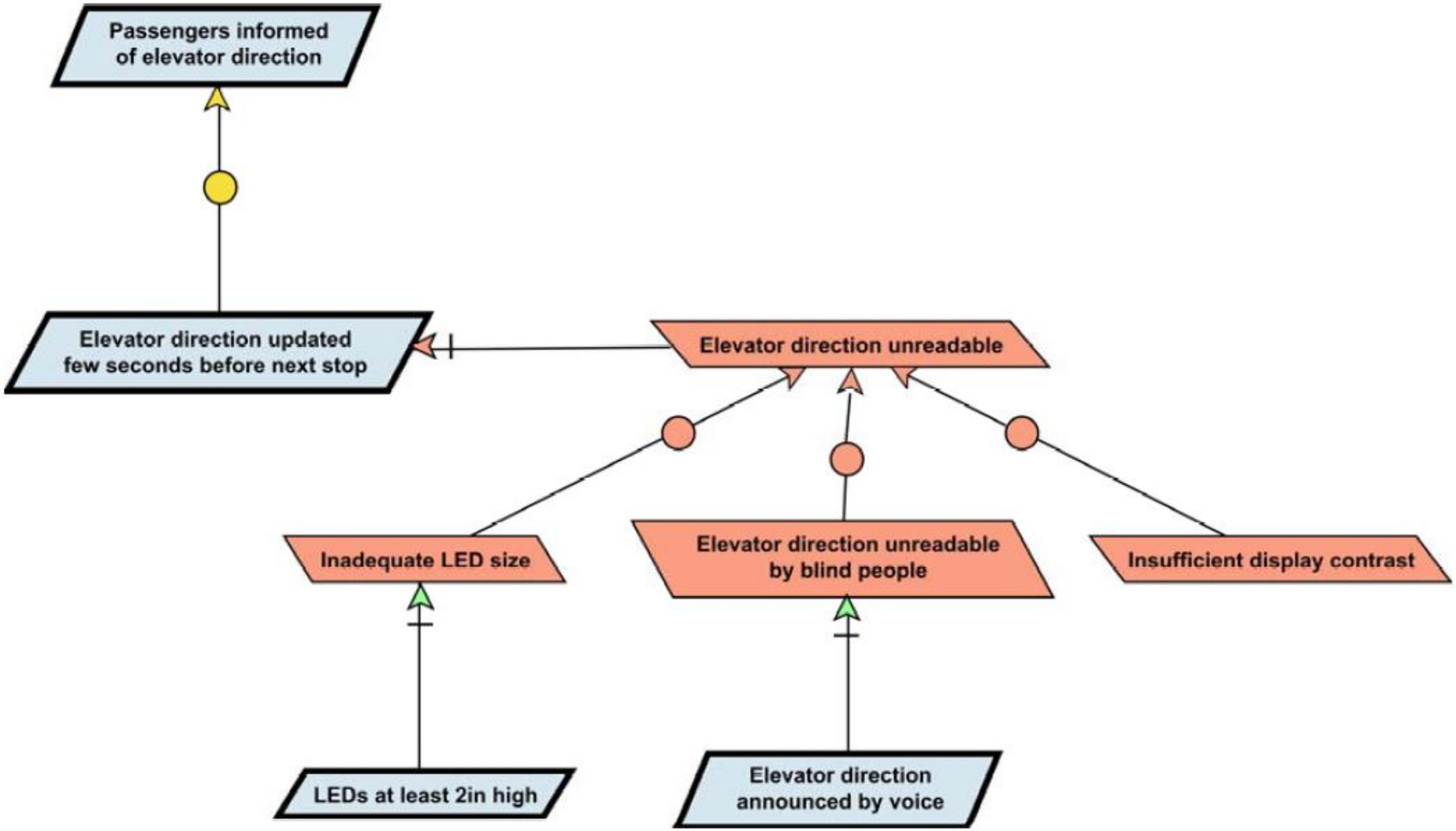
Completeness of Process Diags.

- ***Completeness criterion 3:*** *To be complete, a process diagram must specify*
  - *(i) the agents who perform the operations*
  - *(ii) the input and output data for each operation.*
- ***Completeness criterion 4:*** *To be complete, a process diagram must specify when operations are to be executed.*
- ***Completeness criterion 5:*** *All operations are to be justified by the existence of some requirements (through the use of operationalization links).*

Dealing with obstacles

- Obstacle analysis
  - negating each goal in turn → **obstacle**
  - refine obstacle:
    - most often OR-refined (in contrast to goals)
- Add new requirements that **resolve** obstacles

Dealing with obstacles



References

[1] KAOS-Tool Objectiver

Free trial version unter www.objectiver.com

[2] KAOS tutorial:
http://objectiver.com/fileadmin/download/documents/KaosTutorial.pdf

[3] Presentation “Goal-Oriented Requirements Engineering:
From System Objectives to UML Models to Precise Software Specifications” by Prof. Axel van Lamsweerde:

http://objectiver.com/fileadmin/download/documents/presentations/KaosCEE-AvL.pdf

Remark: All pictures used in this document were taken from the KAOS tutorial.