Case Study: Multi-storey car park

You are acting as a requirements engineering consultant to a client who wants to automate his existing multi-storey car park with time-stamped ticket-issuing machines, payment machines, closed-circuit television cameras in order to deter both theft and non-payment, and automatic barriers operated by

Question:

what are the elements related
*- to the customer **problem***
*- to the **solution** he has in mind?*

Without goal orientation...

Request Button

Barrier

Ticket Issuing Machine (TIM)

Car Park Management System

Given

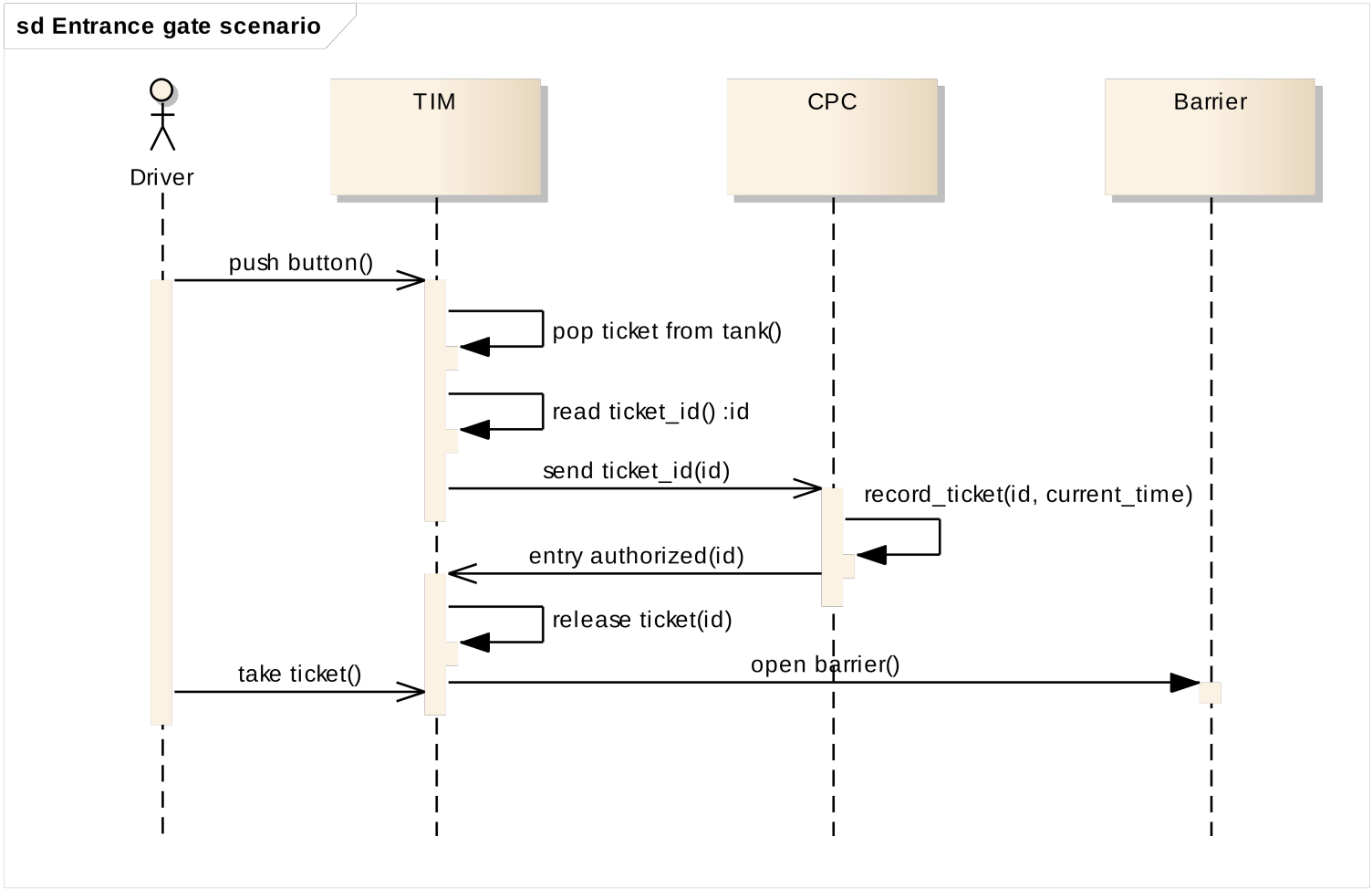
Car in front of TIM

When

Car driver pushes on *entry request* button

Then

System delivers one ticket and opens barrier



Verification: the specification complies with the scenario
Validation: Is this behaviour what the stakeholders want?

First Goal Identification

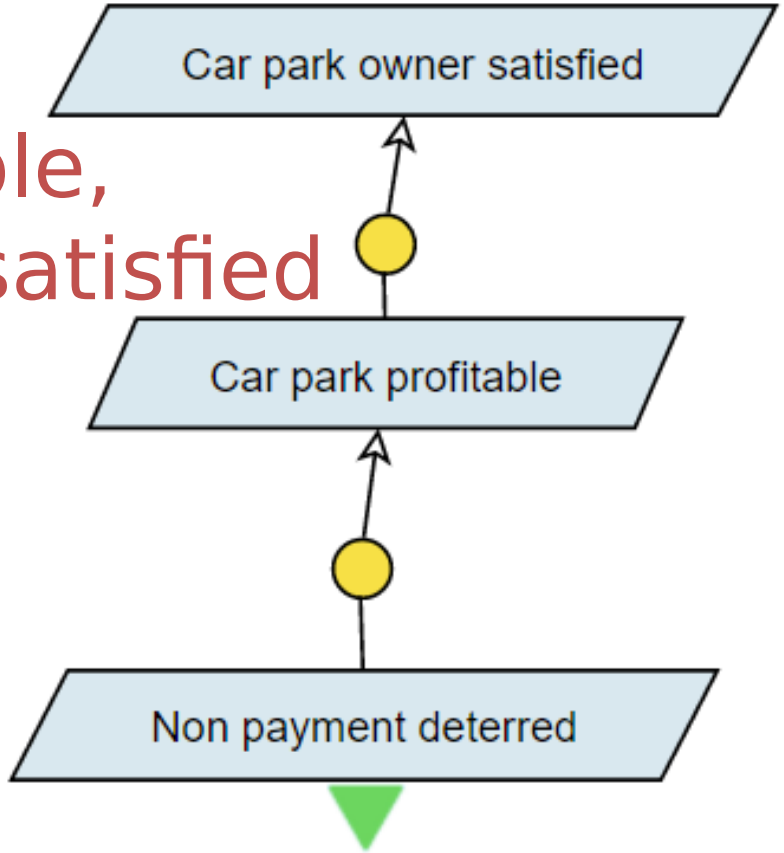
You are acting as requirements engineering consultant to a client who wants to automate his existing multi-storey car park with time-stamped ticket-issuing machines, payment machines, closed-circuit television cameras in order to deter both theft and non-payment, and automatic barriers operated by validated (paid-up) tickets.

- Candidate goals : the problem to solve
- Technical elements: contraints on the solution
- Typical: lots of elts related to a solution
- RE objective: get a deeper insight into the PROBLEM

WHY

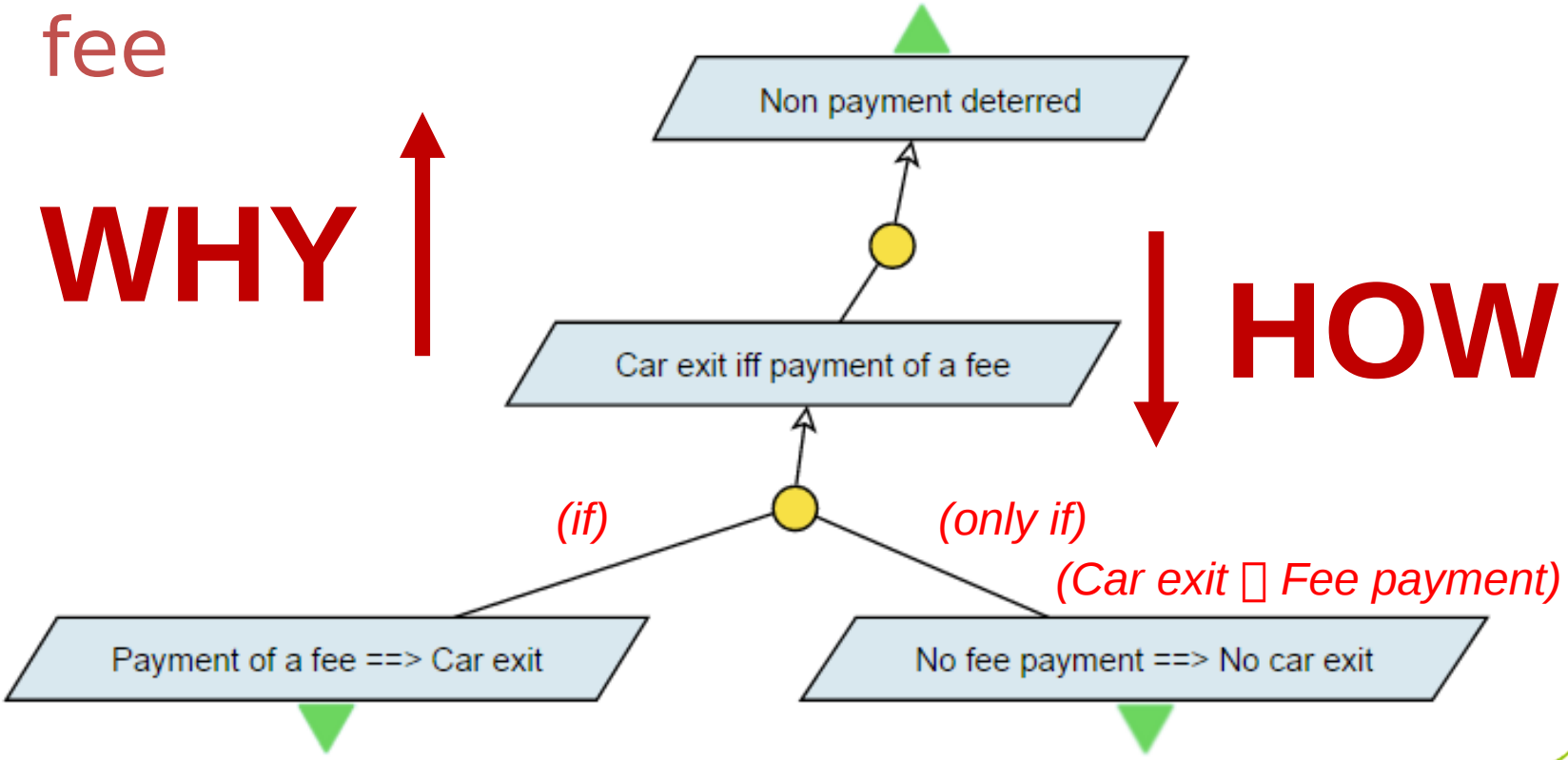
- *Non payment deterred* : **WHY?**
- New goals:
  Car park profitable,
  Car park owner satisfied

WHY

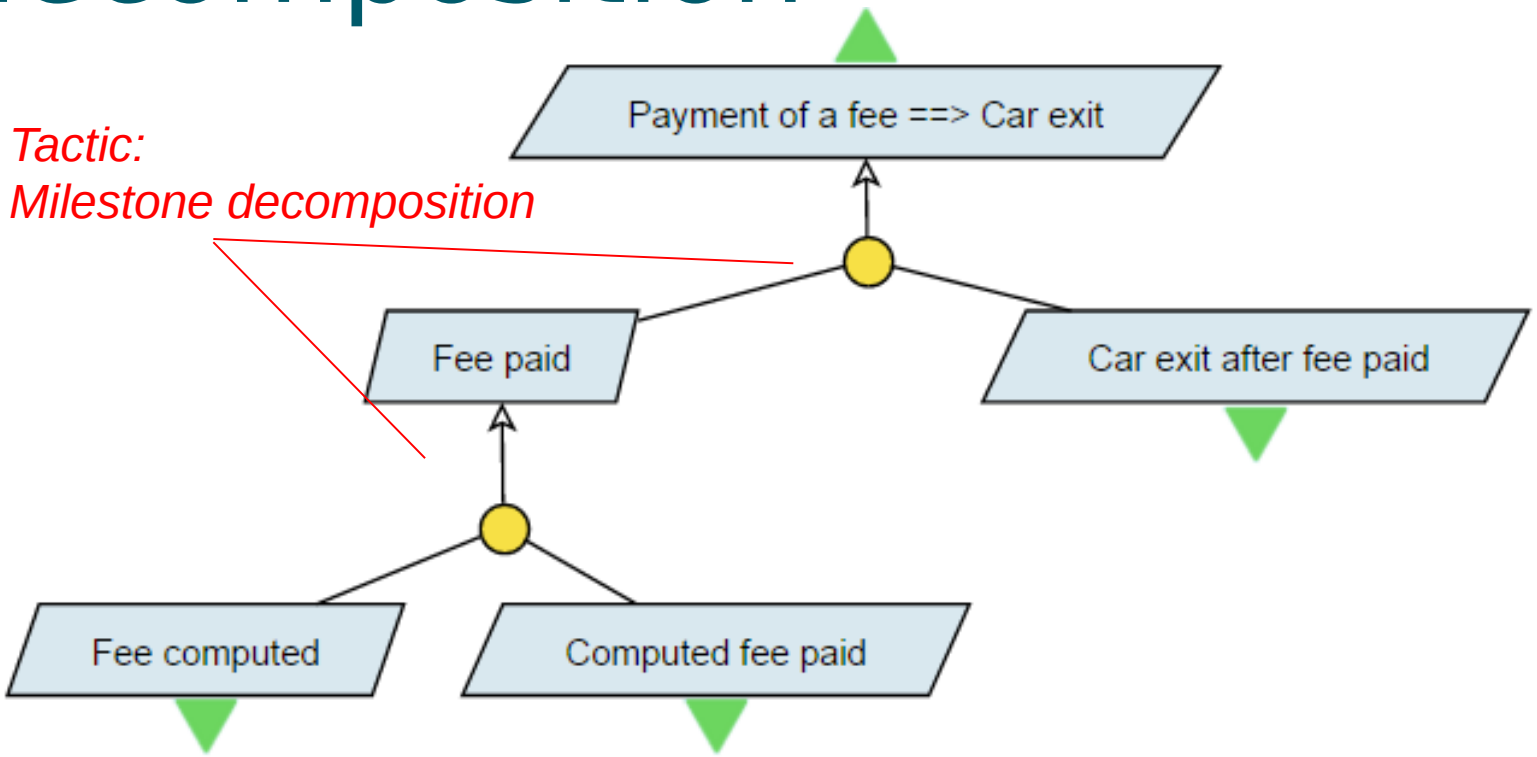


HOW

- *Non payment deterred* : **HOW?**
- New goal: Car exit iff payment of a fee



Tactic: milestone decomposition



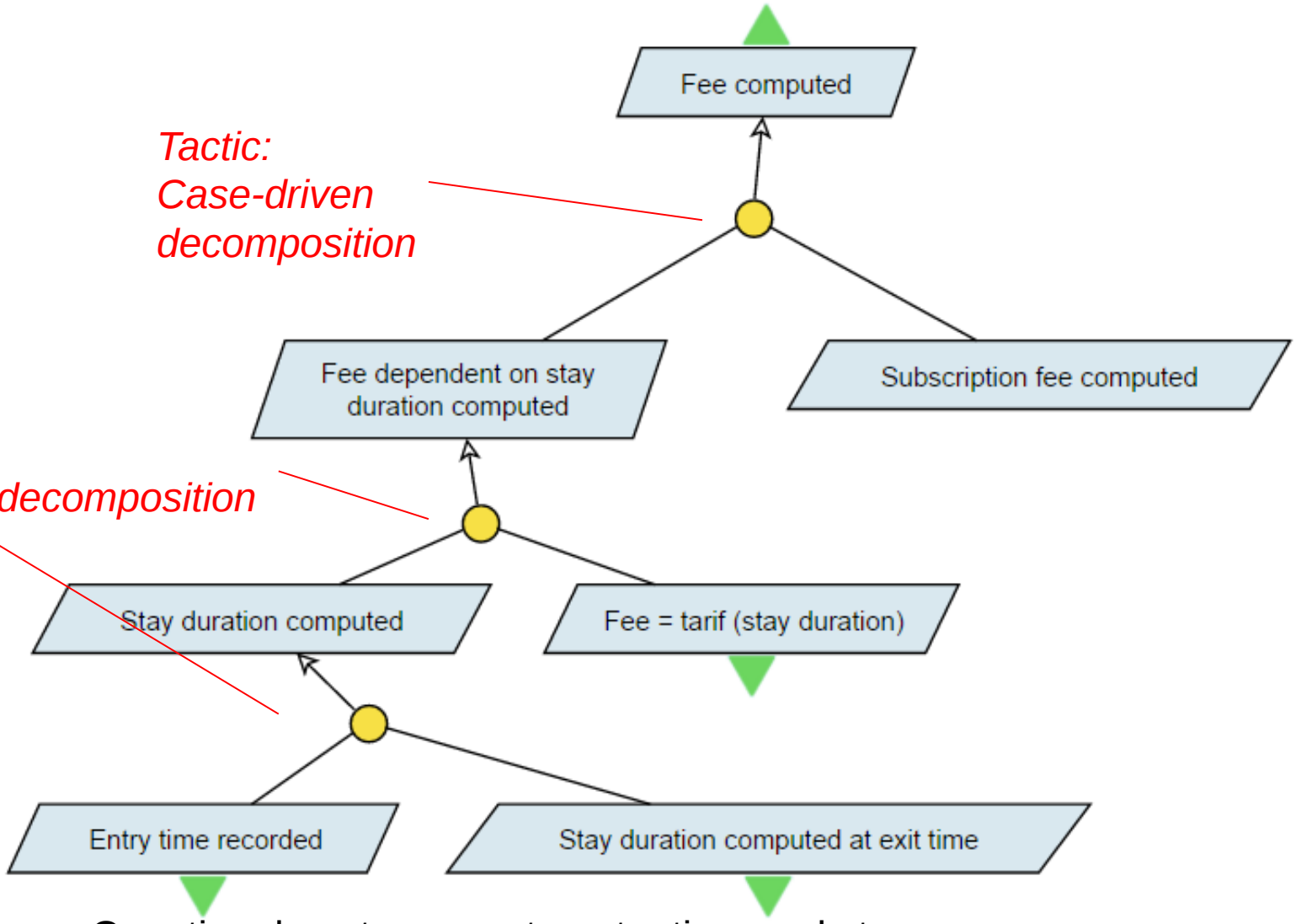
Not yet addressing solution techniques

Just **analysing the pb** based on what's known about the needs

Fee computed

- Several fee policies could be applied:
  - Fee proportional to stay duration (*)
  - Fixed fee
  - Fixed fee per period (*)
  - Fee dependent on user classification
  - …
- Case decomposition tactics
- Applicable cases need to be discussed with the customer.
- Customer selection: (*)

Fee computed



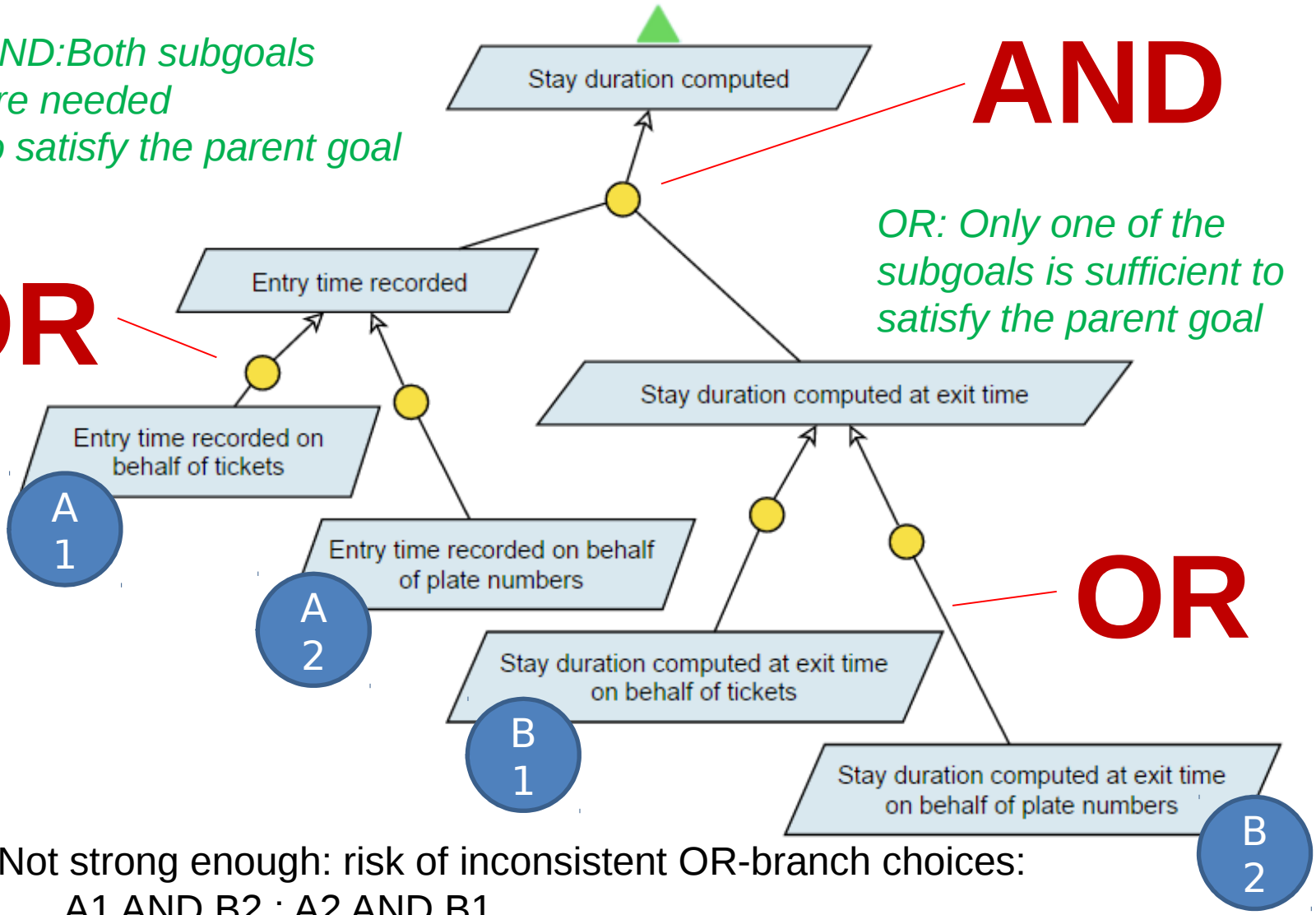
Question: how to compute entry time and stay duration?

Alternatives

- HOW to compute the stay duration?
  - System with tickets
  - System with cameras reading plate numbers
  - Car equipped with id-system
- Goal analysis: allows one to…
  - **envision several solutions** and position them correctly
  - confirm that the expressed needs are real
  - **think about the system**

Alternatives

AND:Both subgoals are needed to satisfy the parent goal

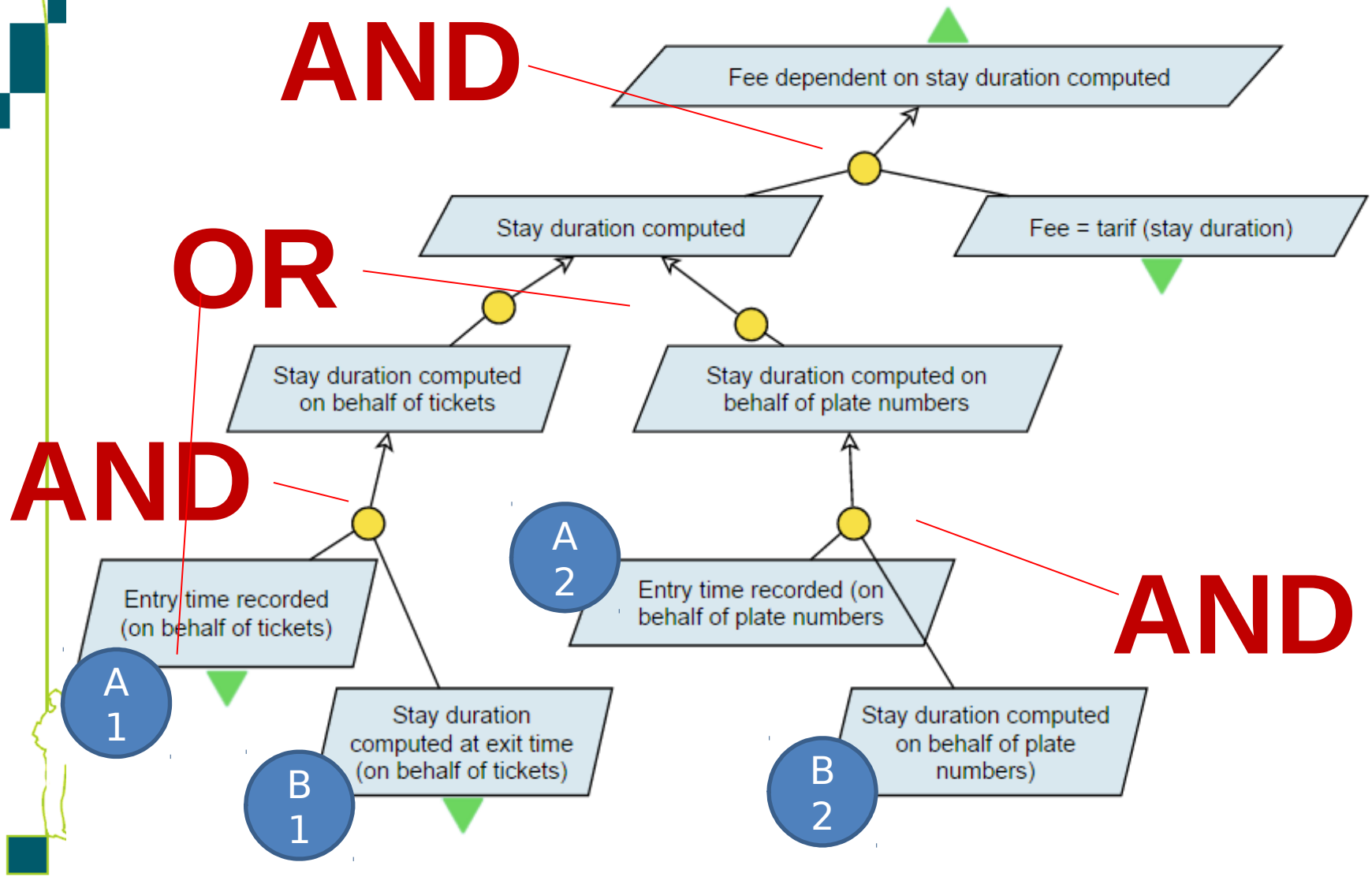


! Not strong enough: risk of inconsistent OR-branch choices:

A1 AND B2 ; A2 AND B1

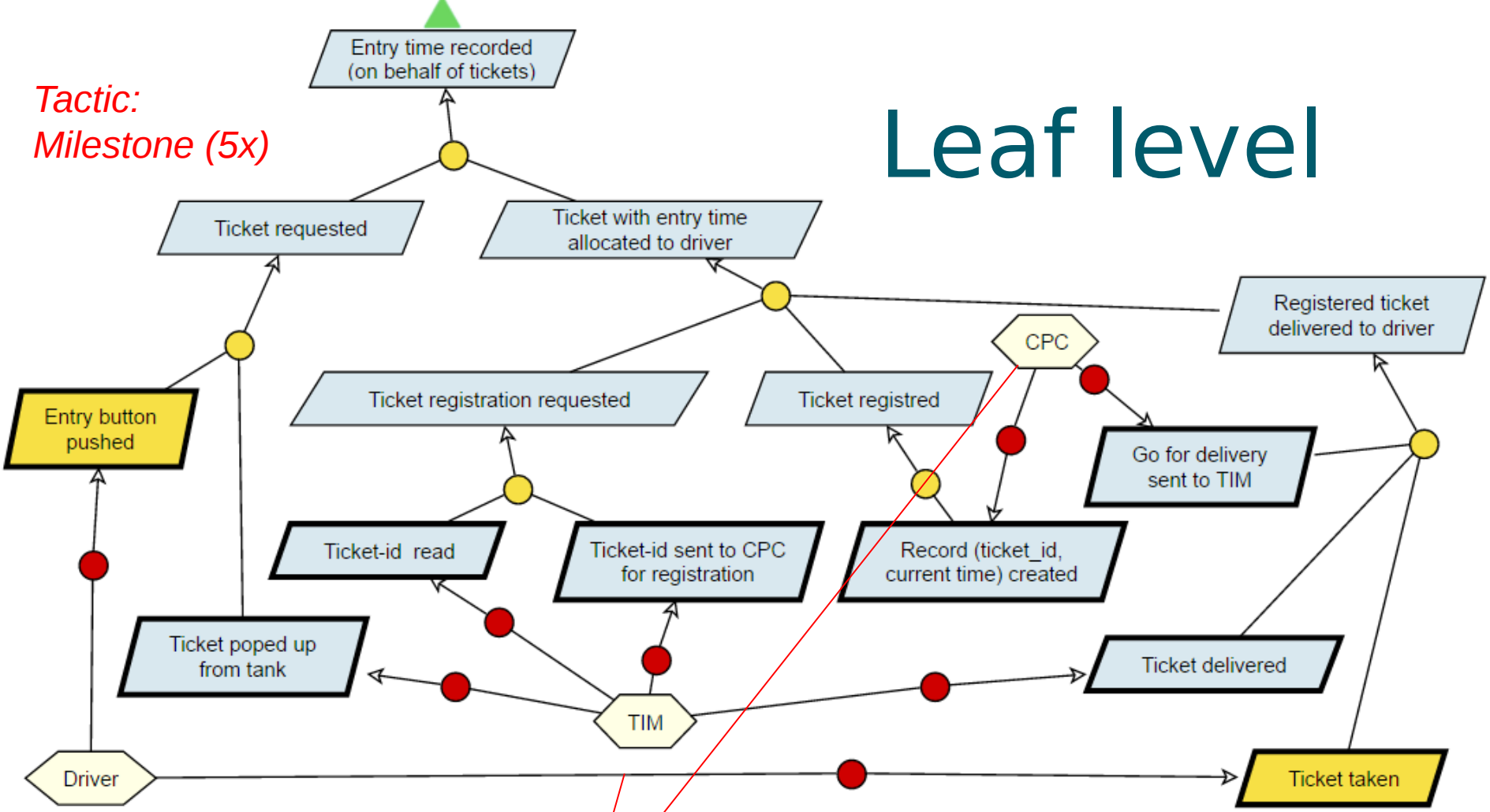
We need (A1 AND B1) OR (A2 AND B2)

After restructuring...



Leaf level

Tactic:
Milestone (5x)



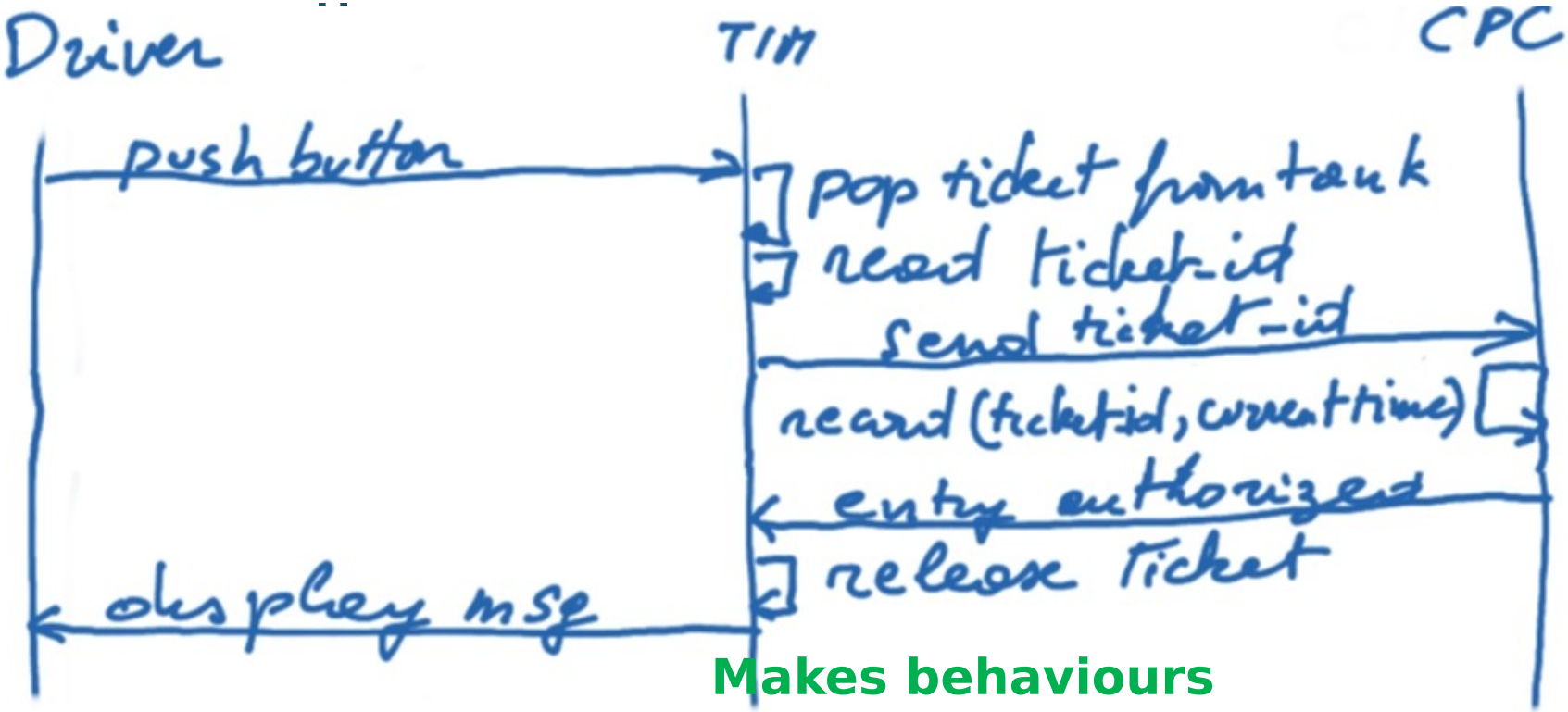
Need to intoduce new agents to operate the missing capabilities…

TIM: Ticket Issuing Machine

CPC: Car Park Controller

Behavioural specification

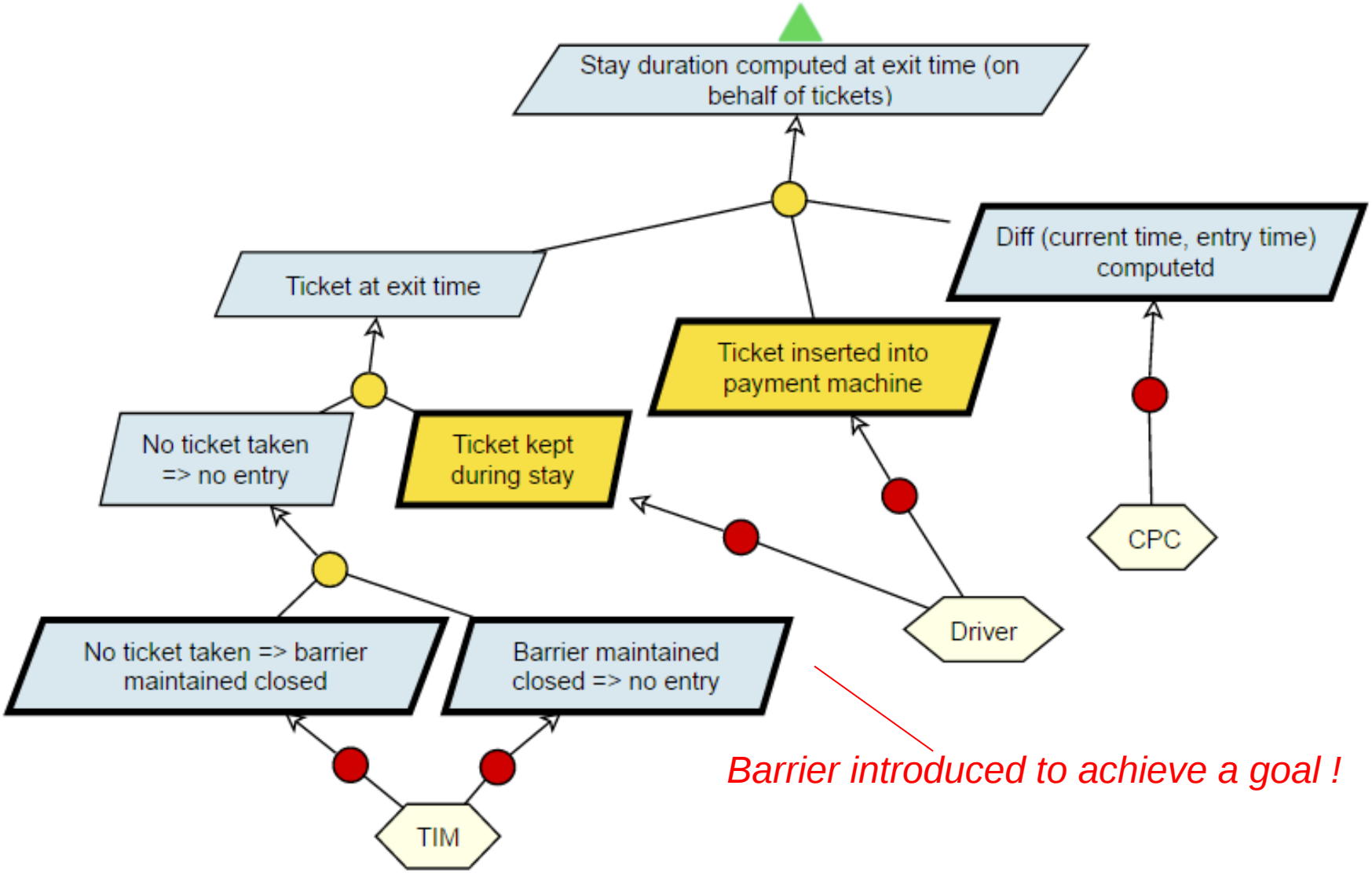
UML sequence or activity

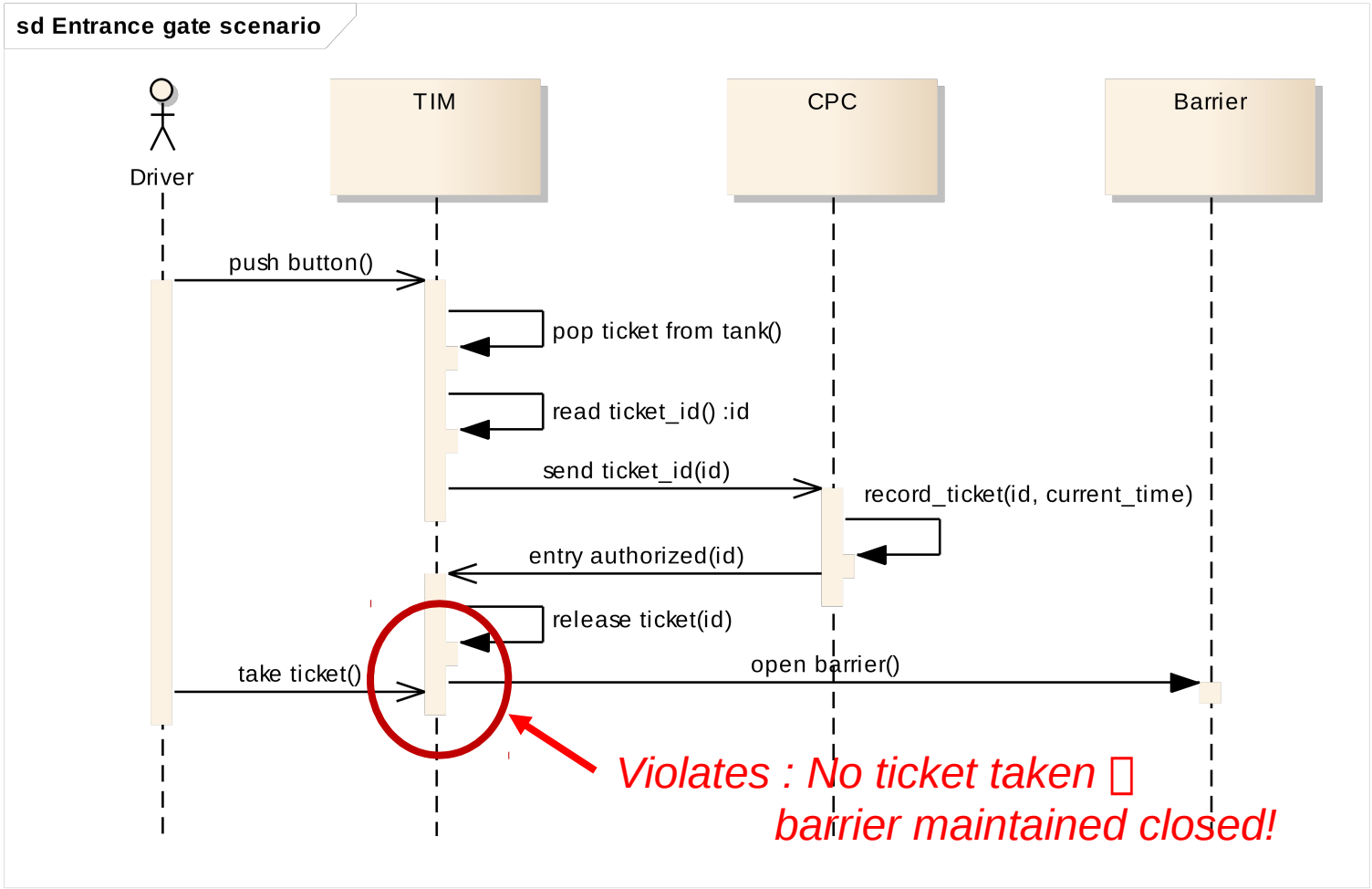


Easy translation from leaf goal diagrams

Makes behaviours more/fully deterministic wrt goals (no order on AND-refinement)

Leaf level


Stay duration computed at exit time (on behalf of tickets)
Ticket at exit time
Diff (current time, entry time) computetd
Ticket inserted into payment machine
No ticket taken => no entry
Ticket kept during stay
CPC
Driver
No ticket taken => barrier maintained closed
Barrier maintained closed => no entry
TIM
Barrier introduced to achieve a goal !

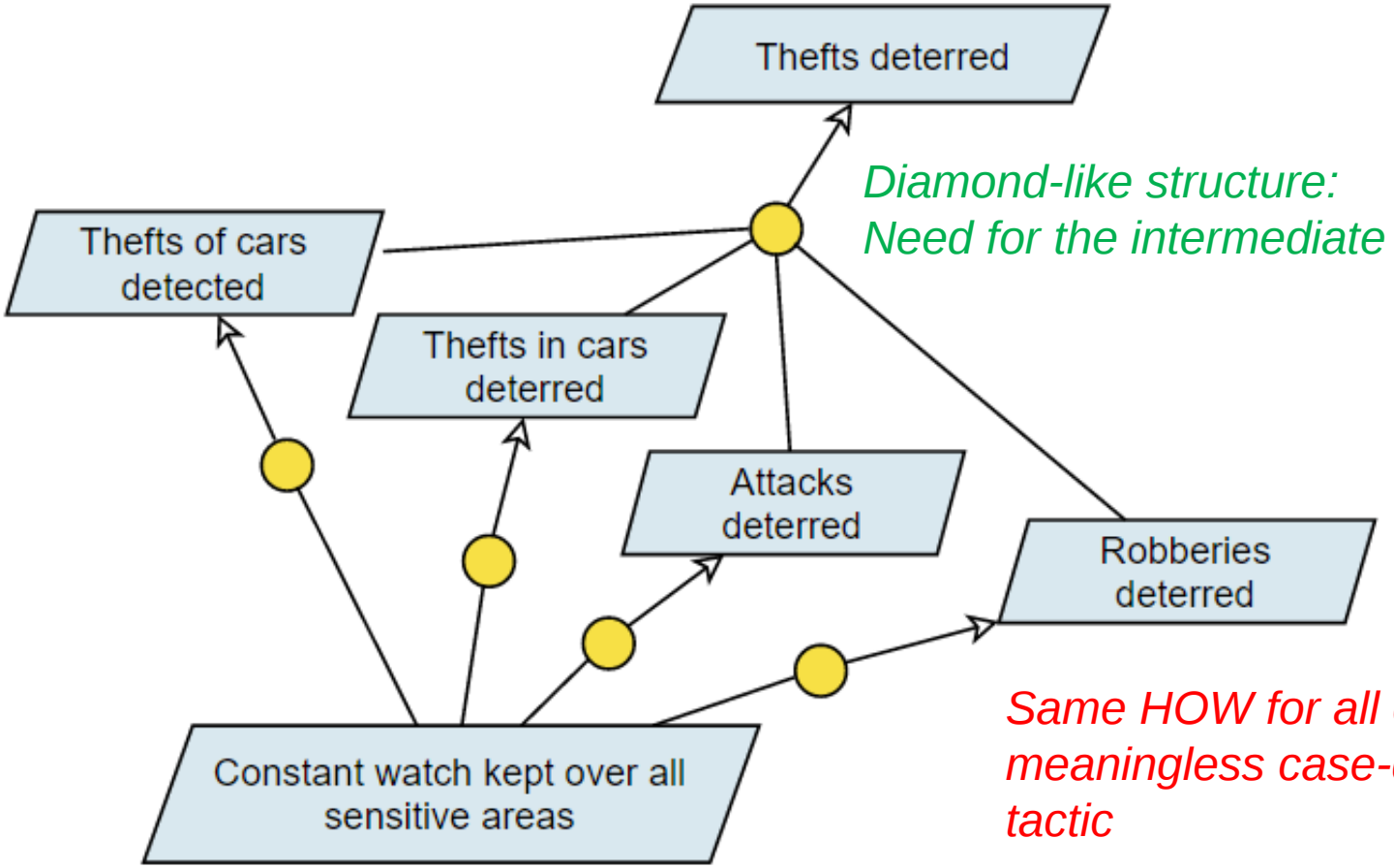


Keypoints:

- Important to keep **rationales** for **behaviours**
- **Traçability**

Thefts deterred

First idea: case decomposition

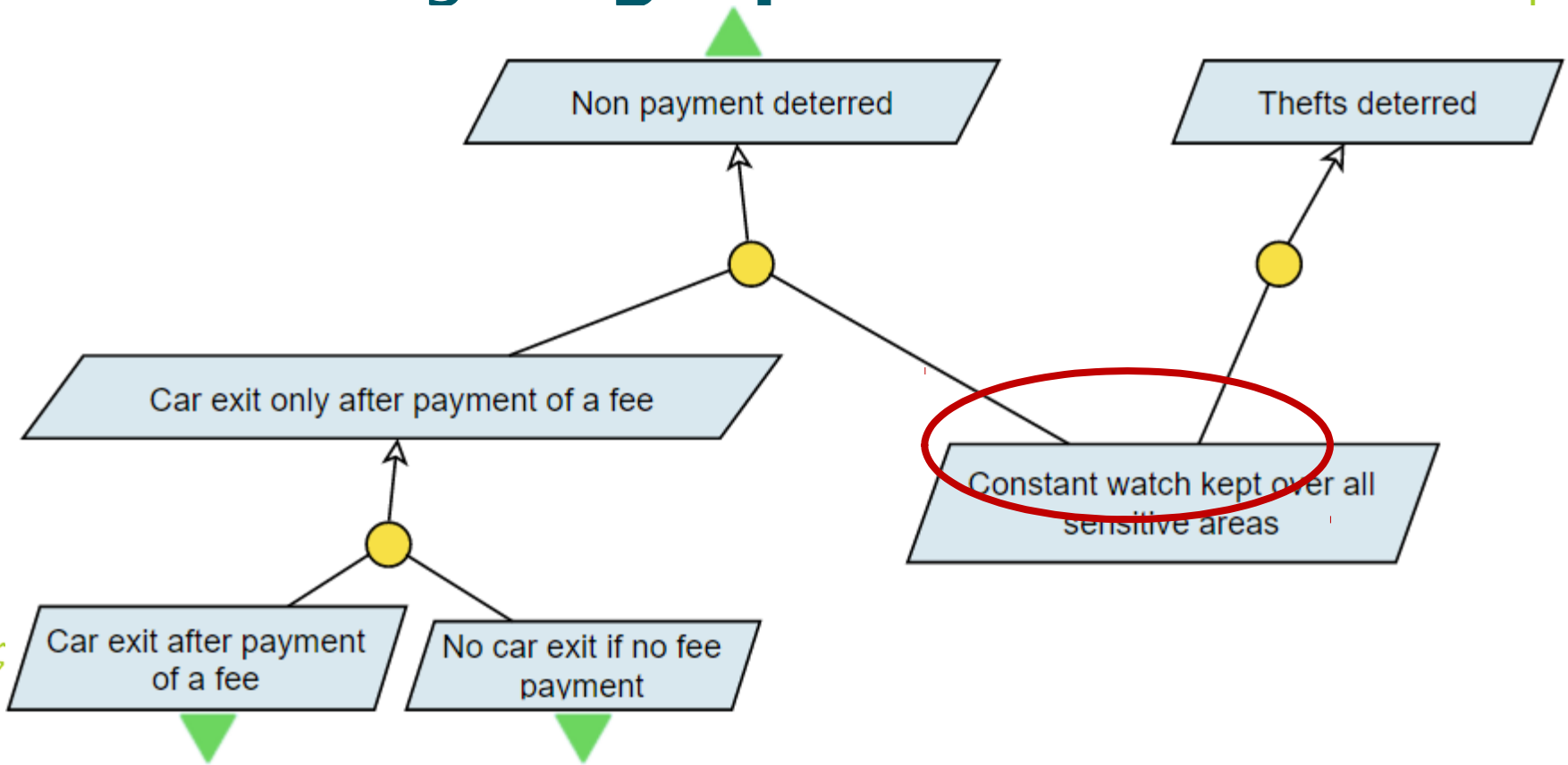


Diamond-like structure:
Need for the intermediate layer???

Same HOW for all cases: meaningless case-driven tactic

Thefts deterred (cont'd)

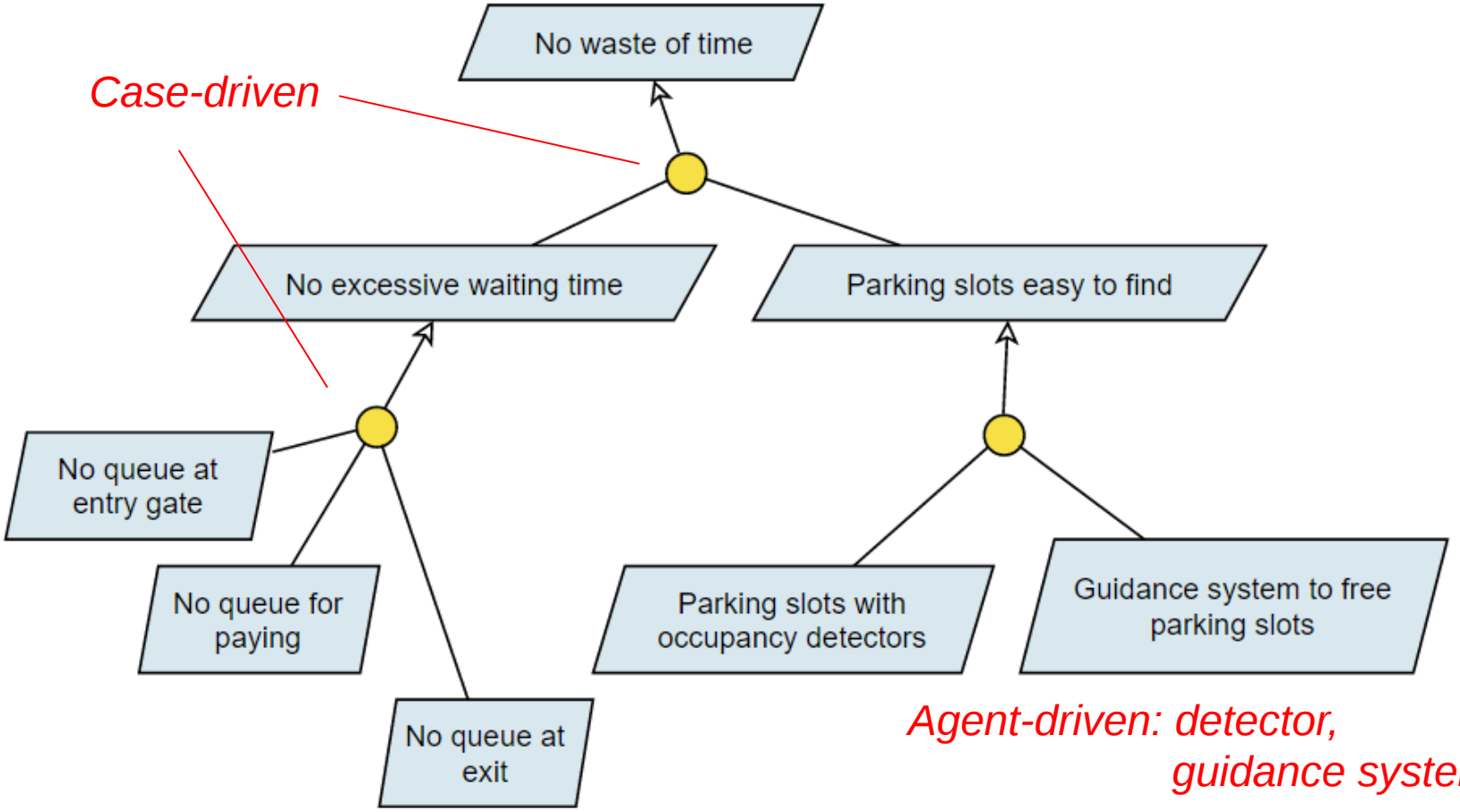
- Directed goal **graph**, not a tree



Take all viewpoints into account

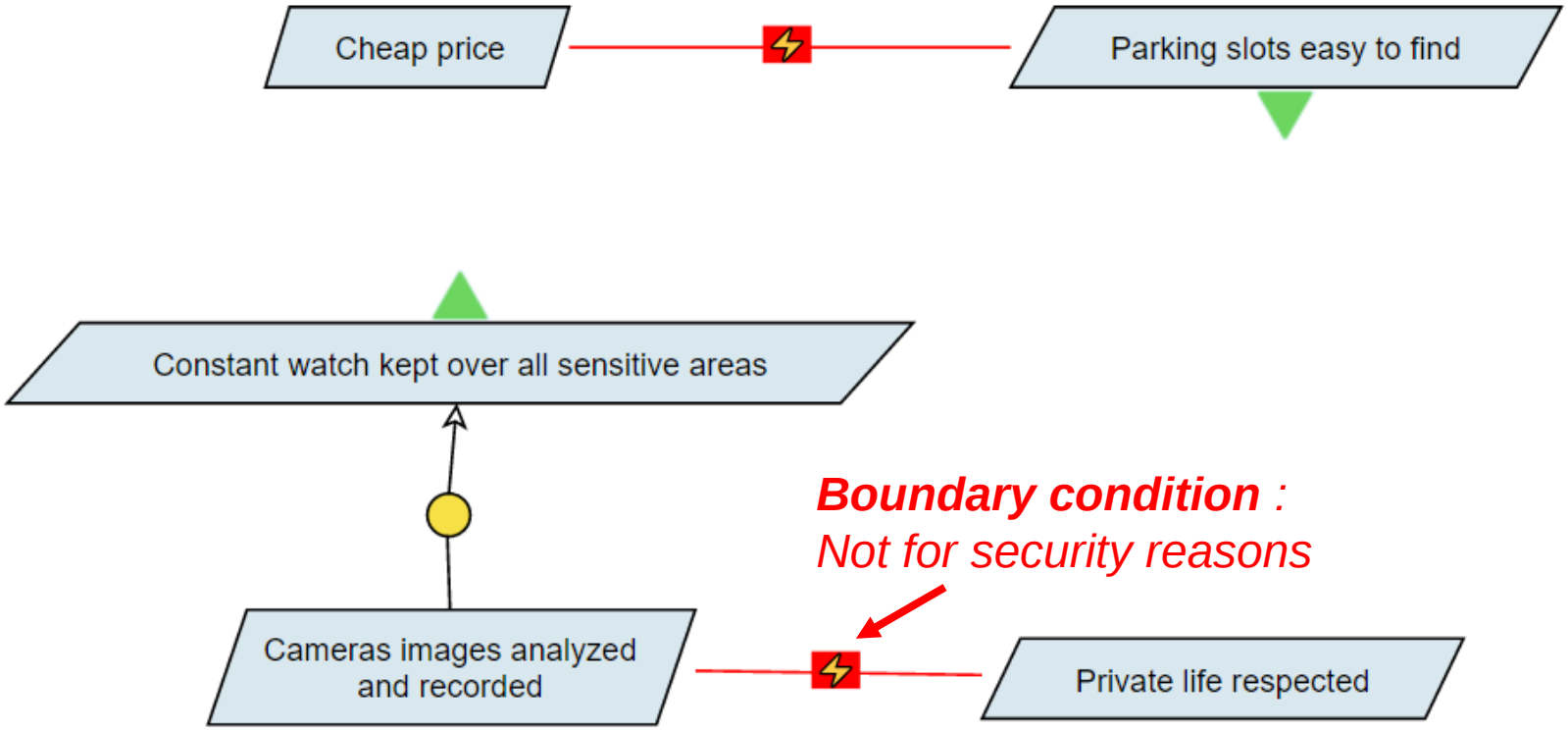
- Viewpoints:
  - Manager
  - Owner
  - User
  - Authorities
- Source for conflicts

Example of user goals



Other goals: cheap price, security, user-friendliness, information goals, ...

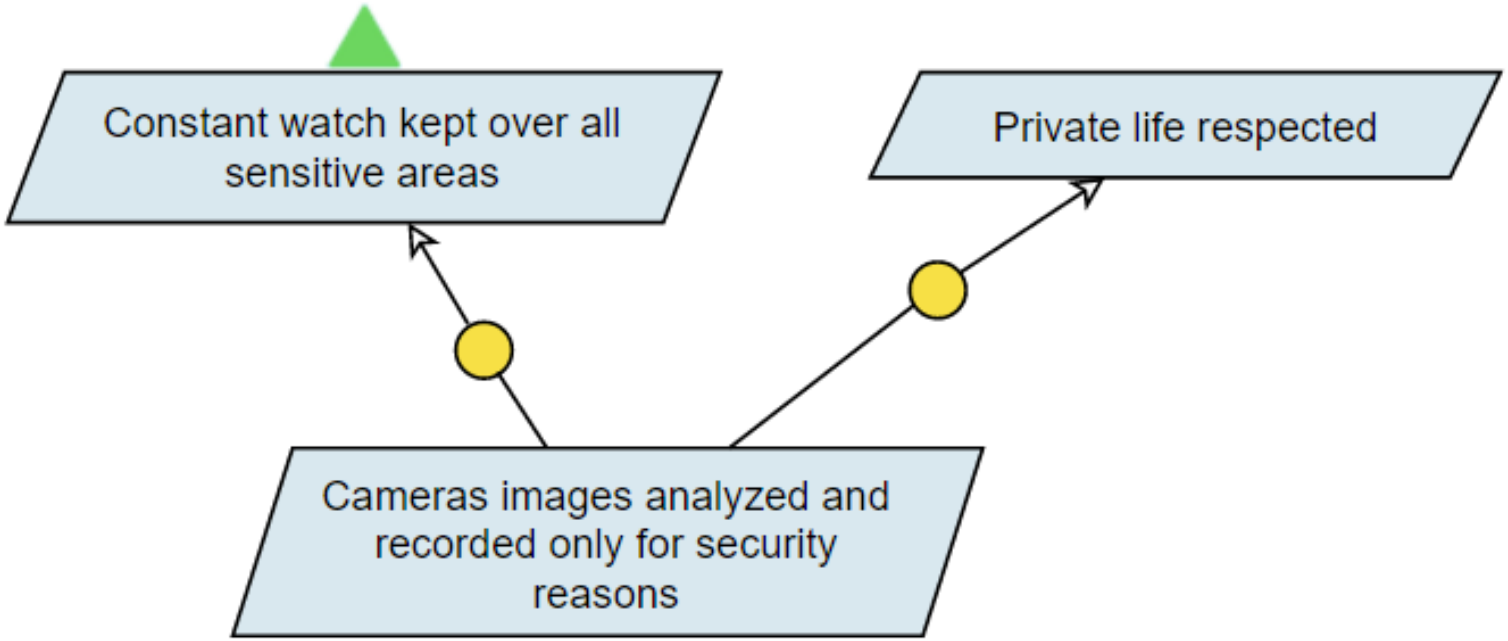
Conflicts



- between goals of a same agent
- between goals of several agents

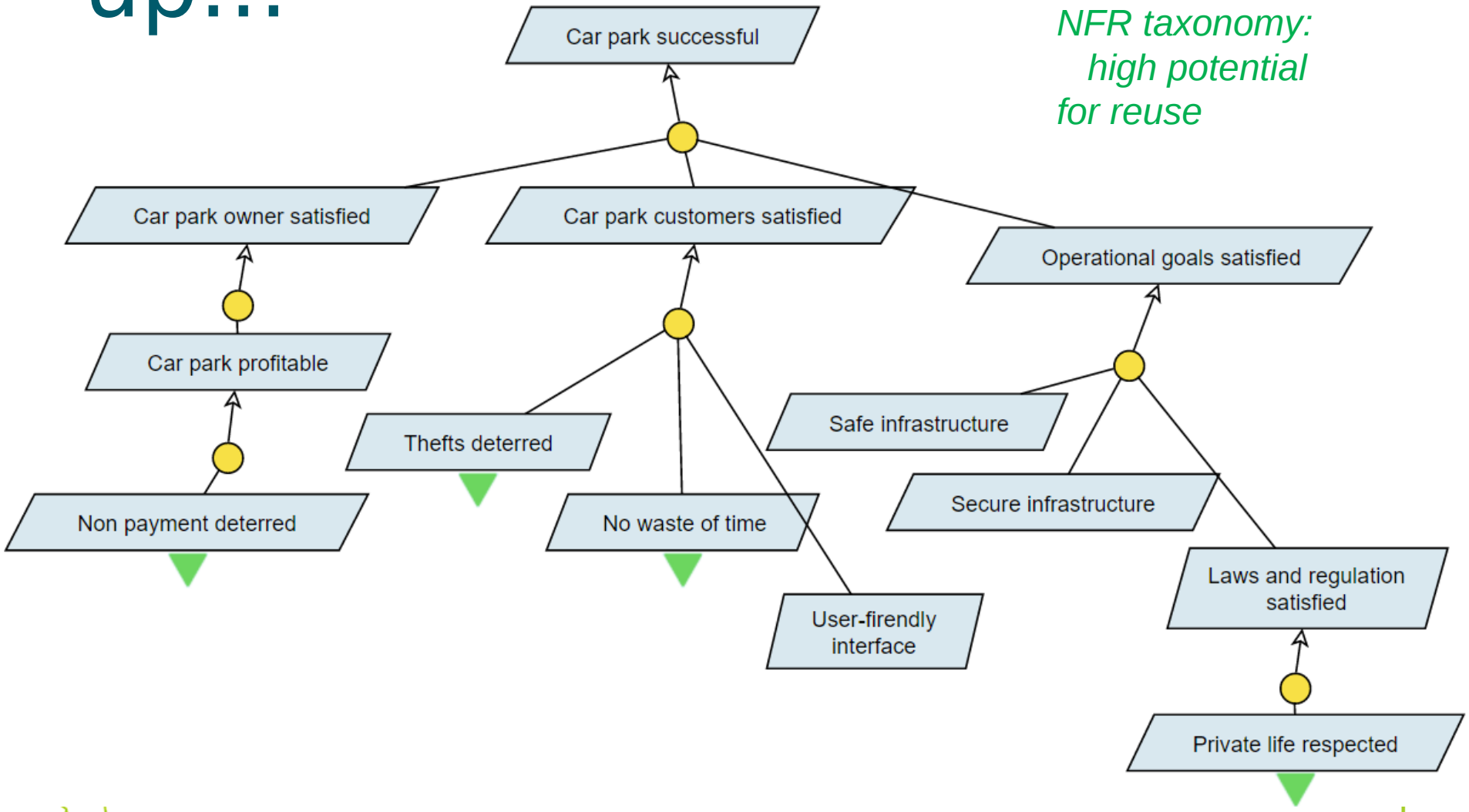
Conflict resolution

- Different techniques, e.g., strengthening conflicting goal to



...or top-down, nor bottom-up...

NFR taxonomy: high potential for reuse

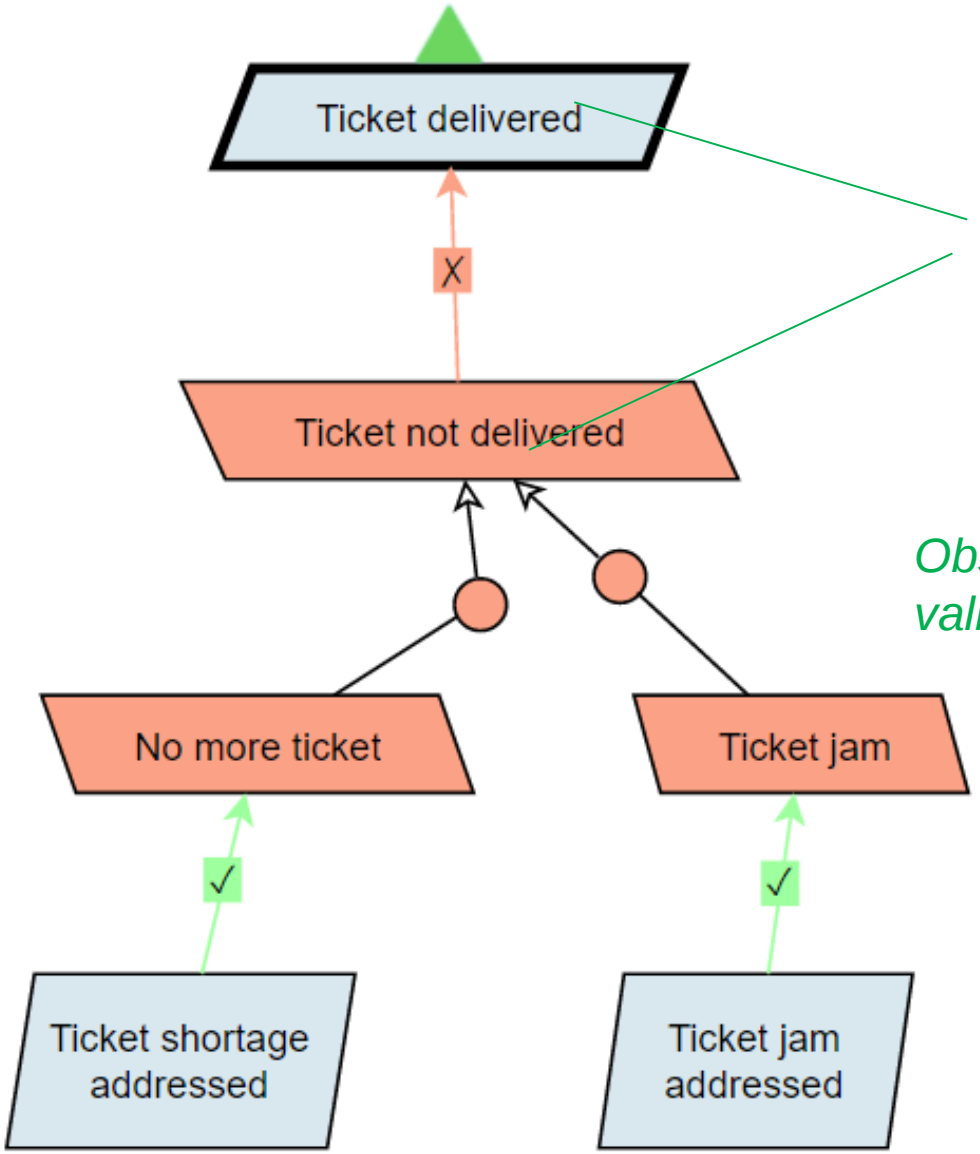


Piecing a jigsaw puzzle together instead...

Obstacles

- De-idealise the model: analyse all what could go wrong
- Process:
  1. Negate the requirement or the expectation
  2. Find the obstacle origin and motivation
  3. Evaluate the obstacle
  4. Solve the obstacle
  5. Integrate in the goal graph

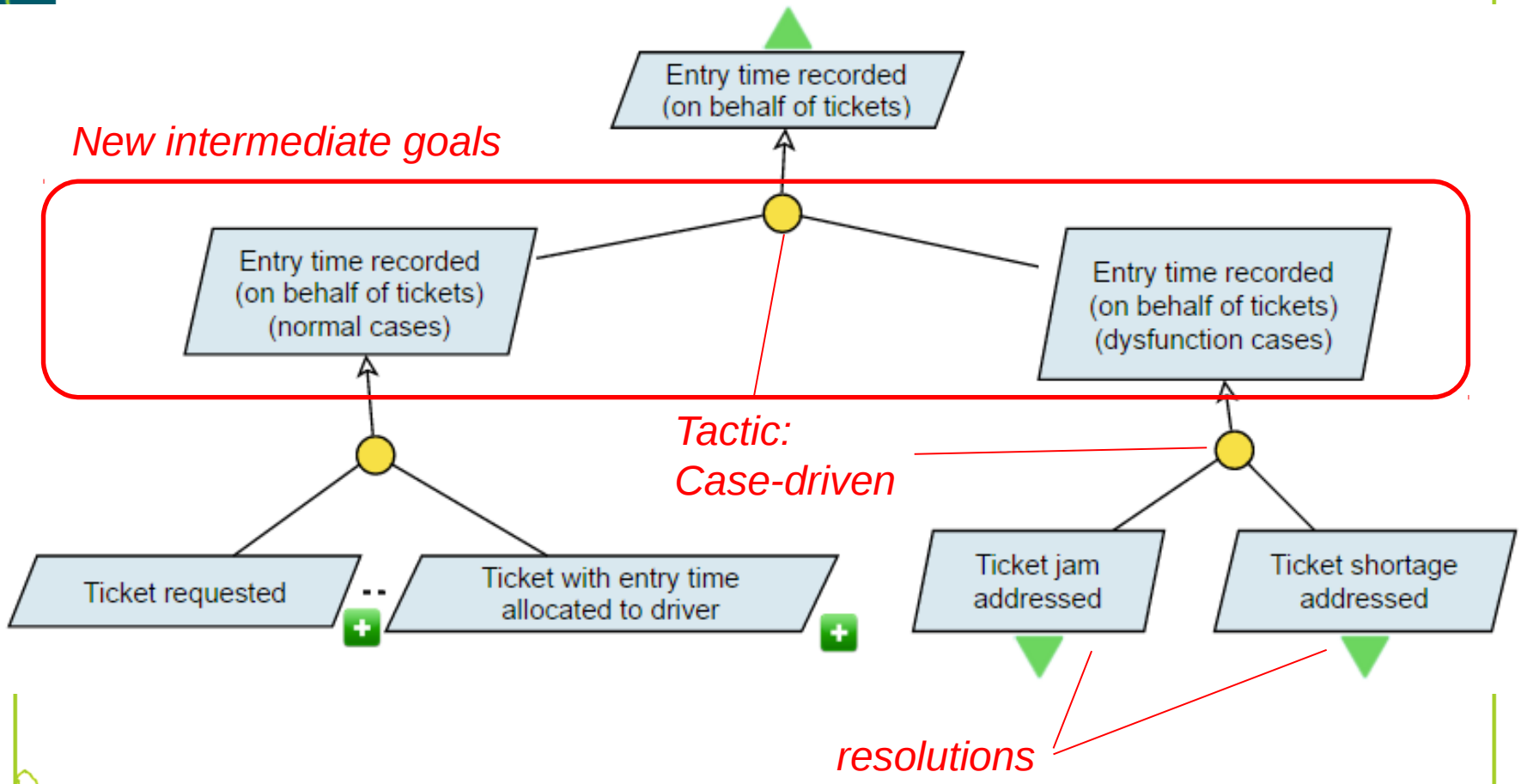
Example



Goal-driven risk analysis

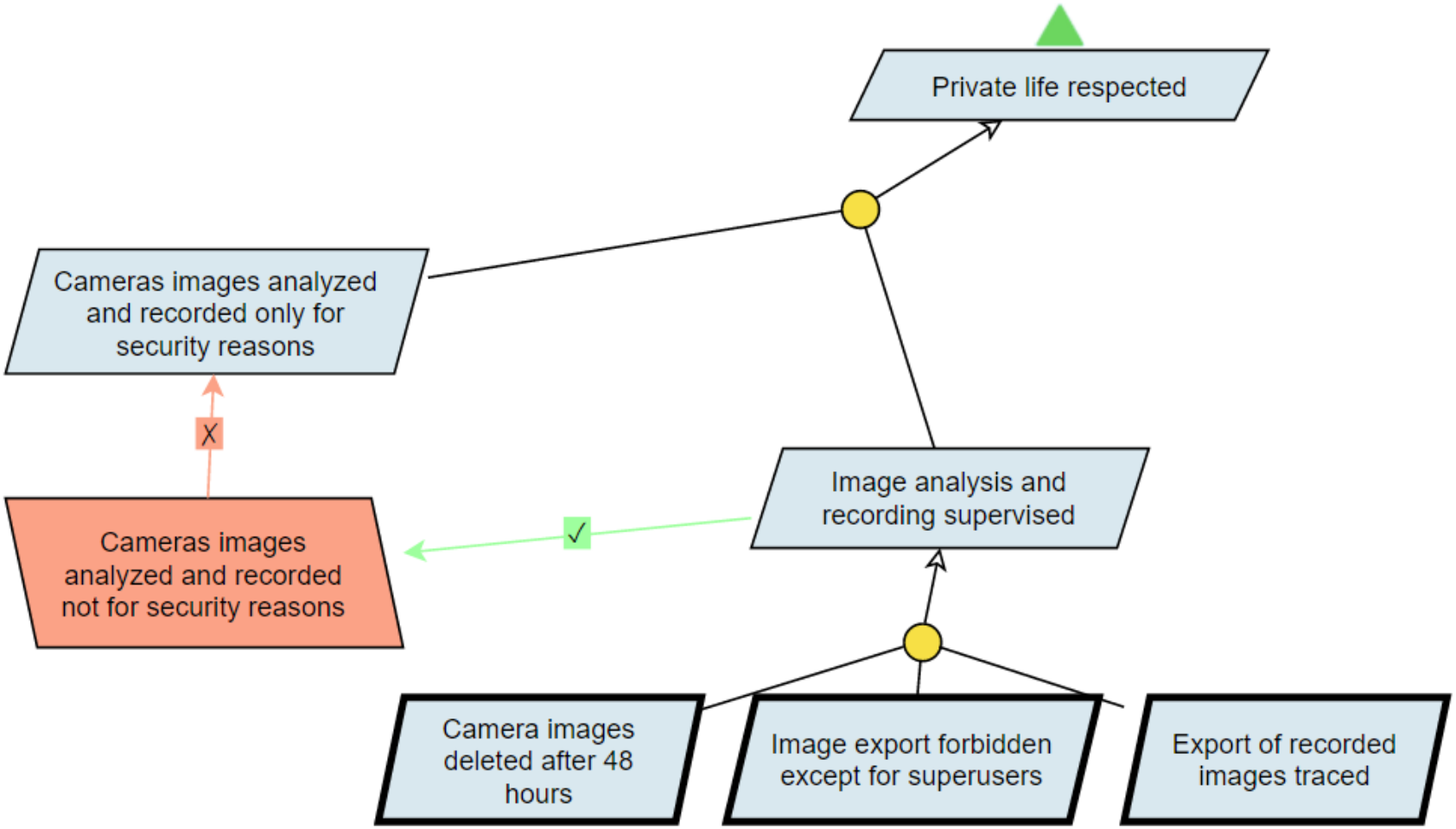
Obstacle pertinence needs validation from domain experts

Example (step 5)



Refinements complemented with obstacle resolution

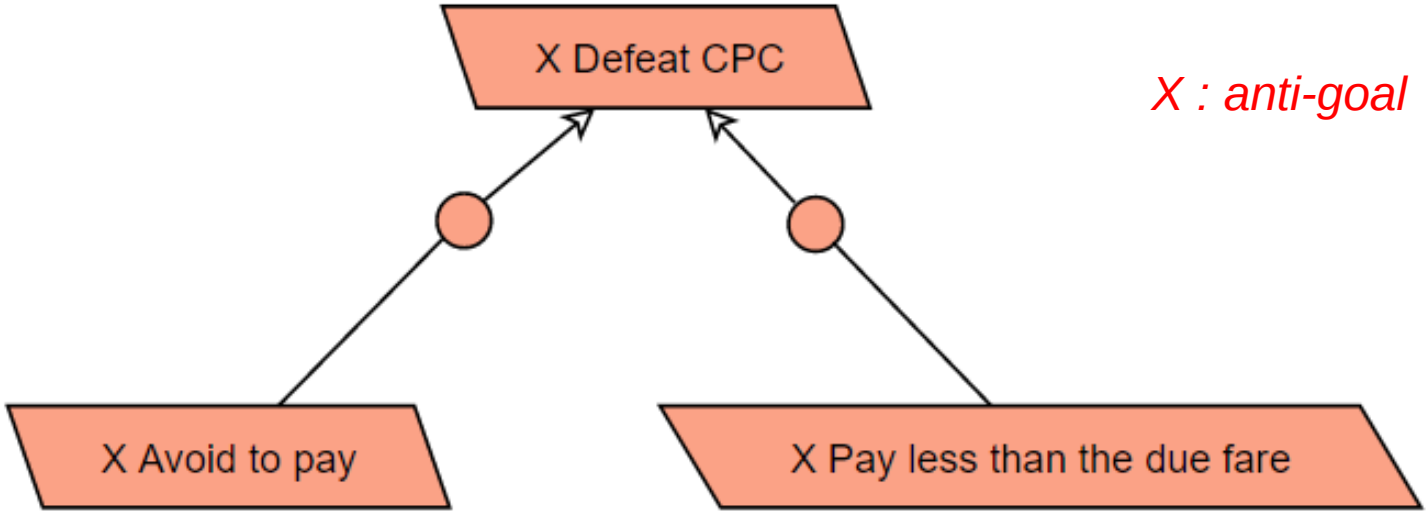
Example



Obstacle resolution: new requirements!

The best defense, ...

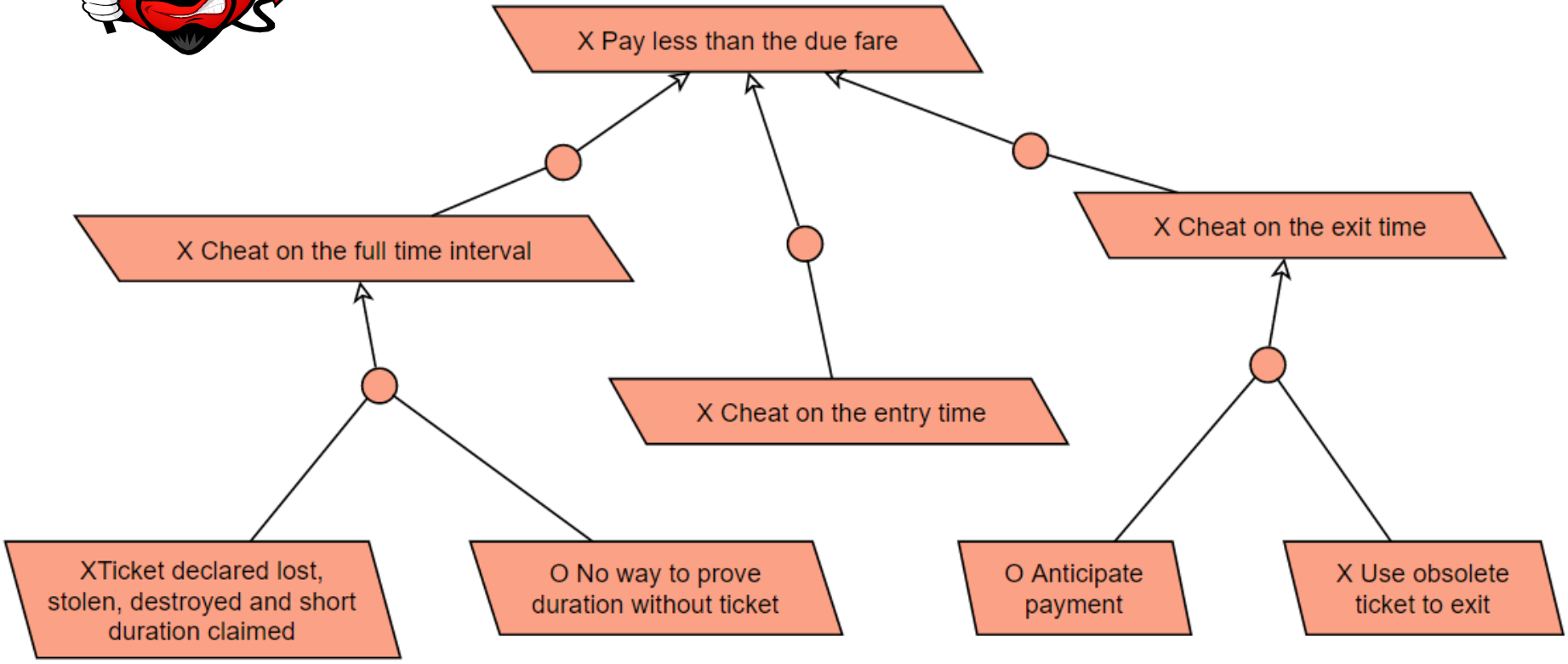
- Playing the bad guy
- Define the anti-goals and refine them into anti-requirements
- Identify the vulnerabilities
- Solve the anti-requirements and the vulnerabilities



X : anti-goal

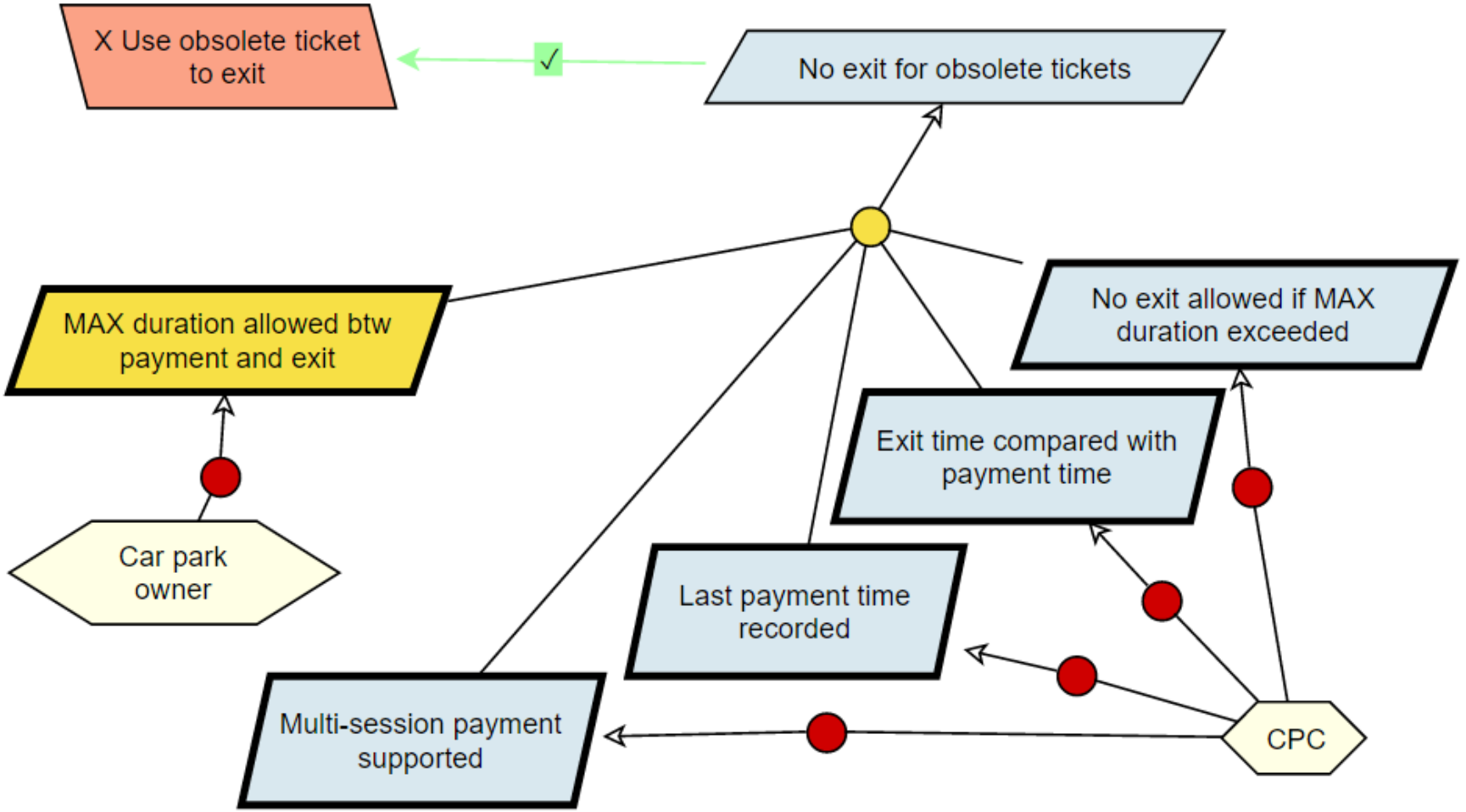
Think negatively...

Vulnerability

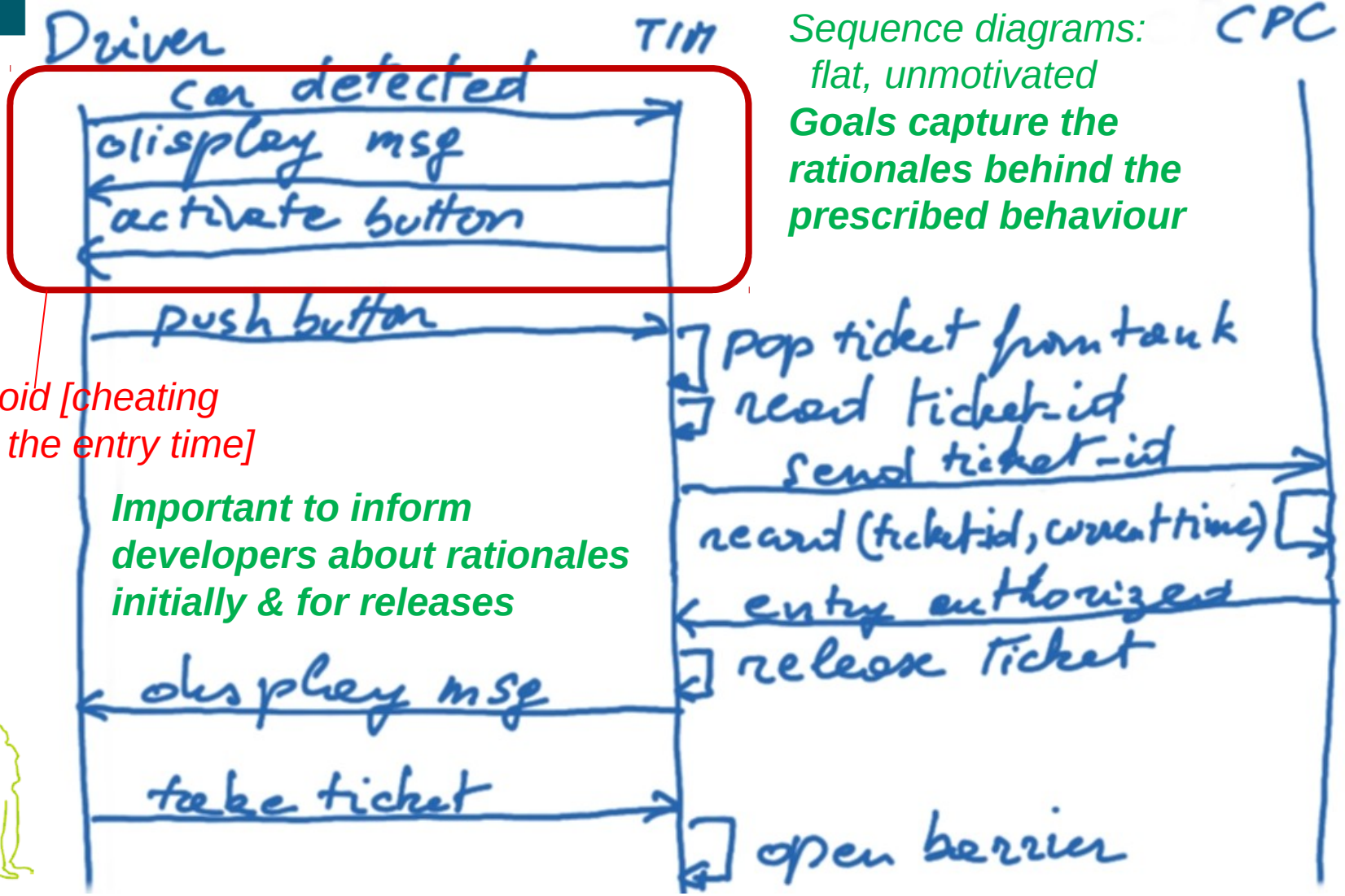


Systematic analysis of delinquent behaviours

Adopt counter-measures



Behavioural Specification



Sequence diagrams: flat, unmotivated

Goals capture the rationales behind the prescribed behaviour

Avoid [cheating on the entry time]

Important to inform developers about rationales initially & for releases

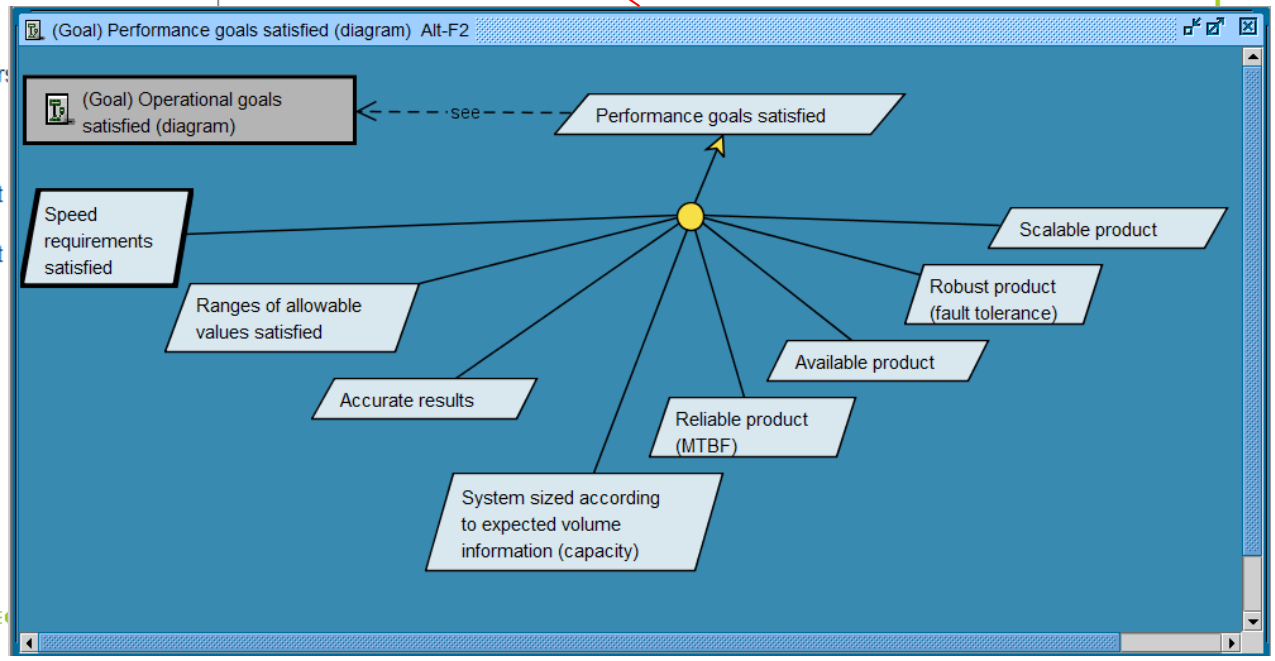
Advanced feature: template

- **Generic** goal graph providing a standardized structure
- To be selected at **project start**
- Covers
  - **NFR** topics (security, reliability, …)
  - Possibly functional goals related to the **project domain** (reused from past projects)
- NFR graph can be used as a **checklist**

Template: extract

- Index de concepts
- Requirements Model
  - NFR
    - Operational goals
      - Performance goals
        - (Goal) Performance goals satisfied (diagram)
      - Security Goals
        - (Goal) Auditable product (diagram)
        - (Goal) Data and code integrity ensured (diagram)
        - (Goal) Privacy goals maintained (diagram)
        - (Goal) Security goals satisfied (diagram)
      - (Goal) Maintainability goals satisfied (diagram)
      - (Goal) Operational conditions satisfied (diagram)
      - (Goal) Operational goals satisfied (diagram)
      - (Goal) Safety goals satisfied (diagram)
    - Usability goals
      - (Goal) Adaptable product (diagram)
      - (Goal) Product easy to learn (diagram)
      - (Goal) Product easy to use (diagram)
      - (Goal) Product usable by all targetted users
      - (Goal) Usability goals satisfied (diagram)
    - (Goal) Cultural goals considered (diagram)
    - (Goal) Legal aspects in force (diagram)
    - (Goal) Look and feel goals taken into account
    - (Goal) NFR goals satisfied (diagram)
  - (Goal) Requirements on the development project
  - (Goal) Top goals

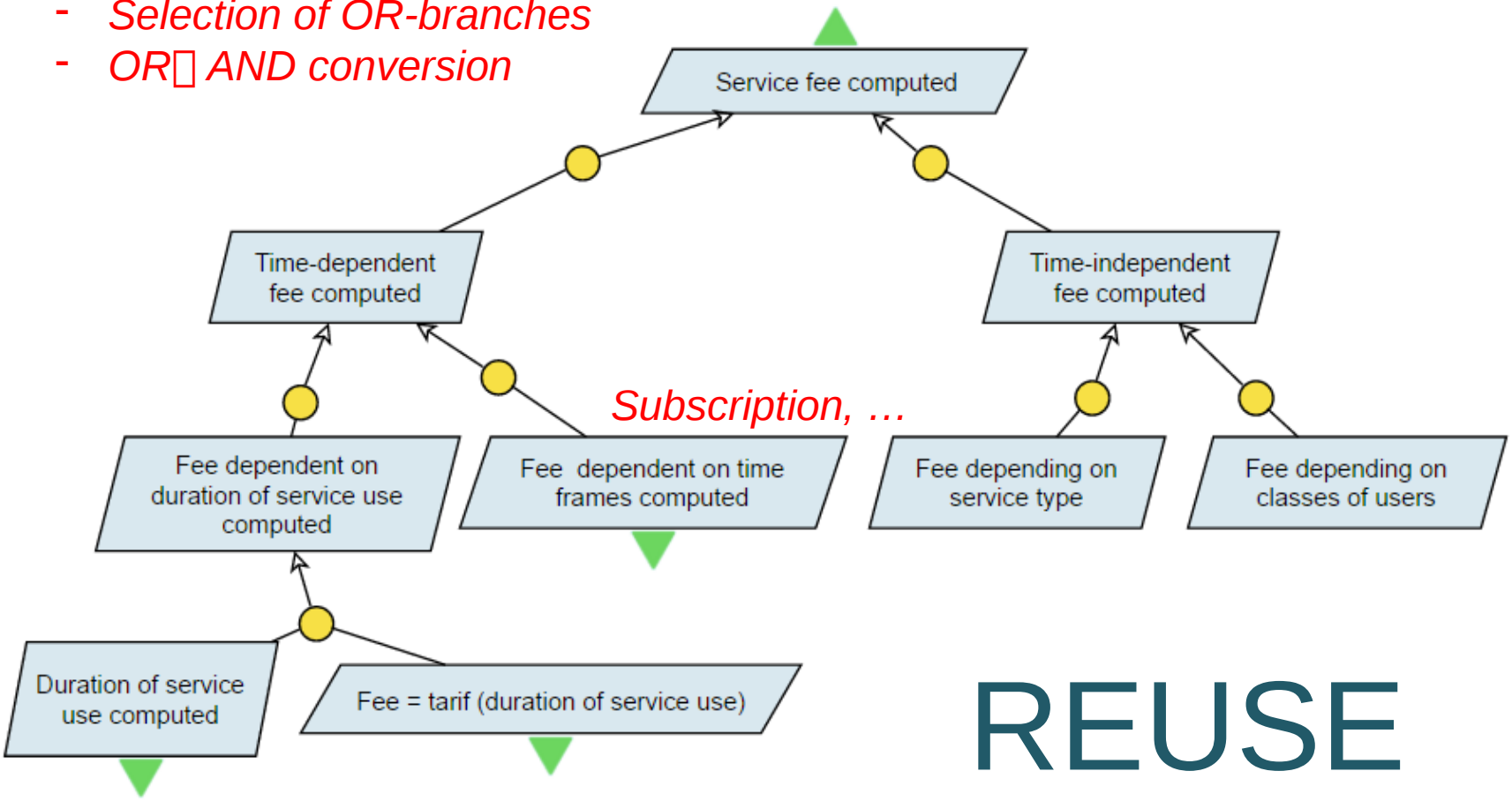
Template structure



Advanced feature : Patterns

Generic piece of goal graph

- *Instantiation to concrete domains*
- *Selection of OR-branches*
- *OR AND conversion*



REUSE

Object Modelling

You are acting as requirements engineering consultant to a client who wants to automate his existing multi-storey car park with time-stamped ticket-issuing machines, payment machines, closed-circuit television cameras to deter both theft and non-payment, and automatic barriers operated by validated (paid-up) tickets.

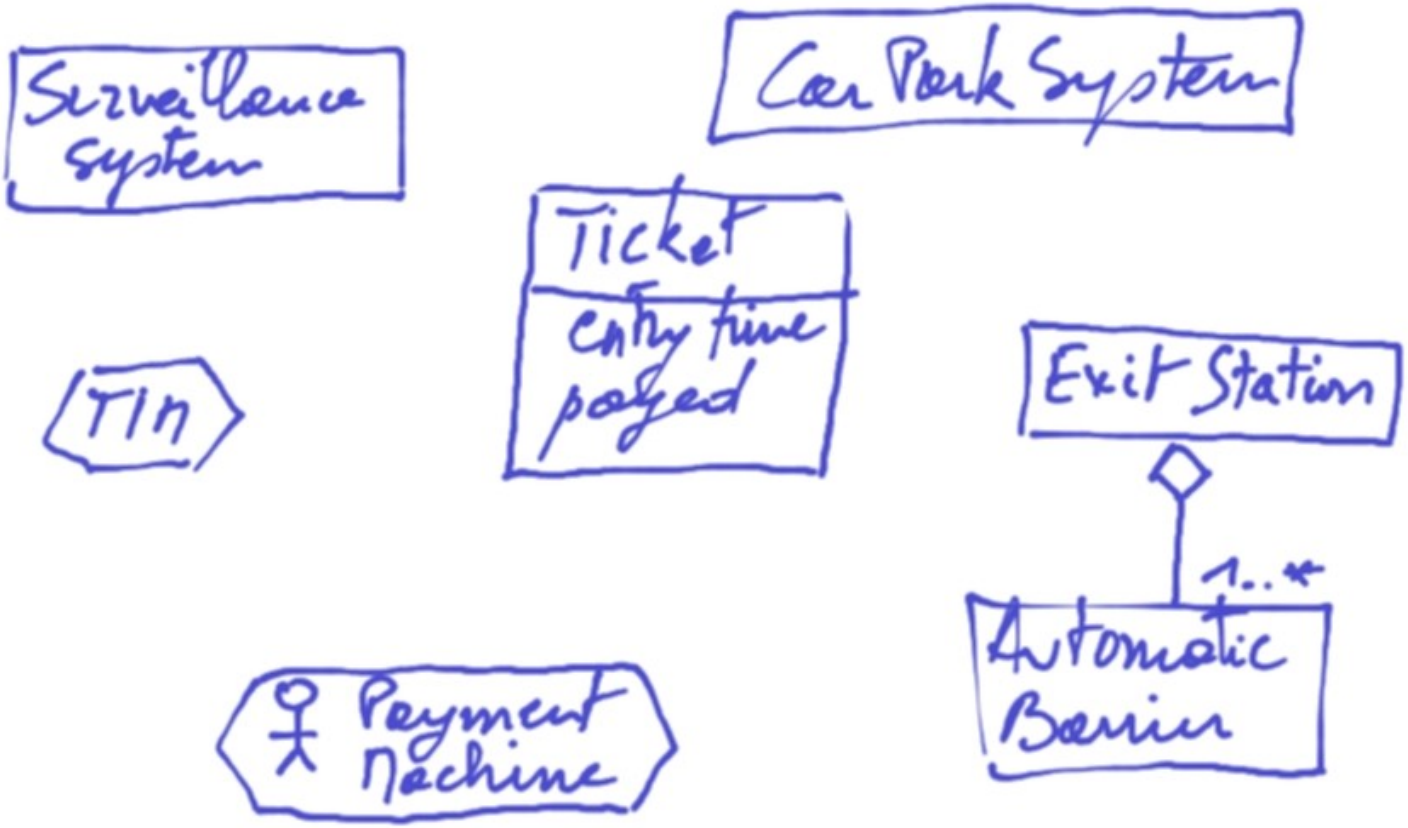
- Candidate objects: domain specific concepts
- Object attributes: object qualification

First object inventory

Statement	Model
Ticket-issuing machine	Ticket Issuing Machine (TIM)
Ticket	Ticket
Payment machine	Payment machine
Closed-circuit television camera	Surveillance system
Automatic barrier operated by validated tickets	Exit station, Automatic barrier
Car park	Car Park System

Ticket	Attribute
Stamped	entry time
Validated (payed up)	payed

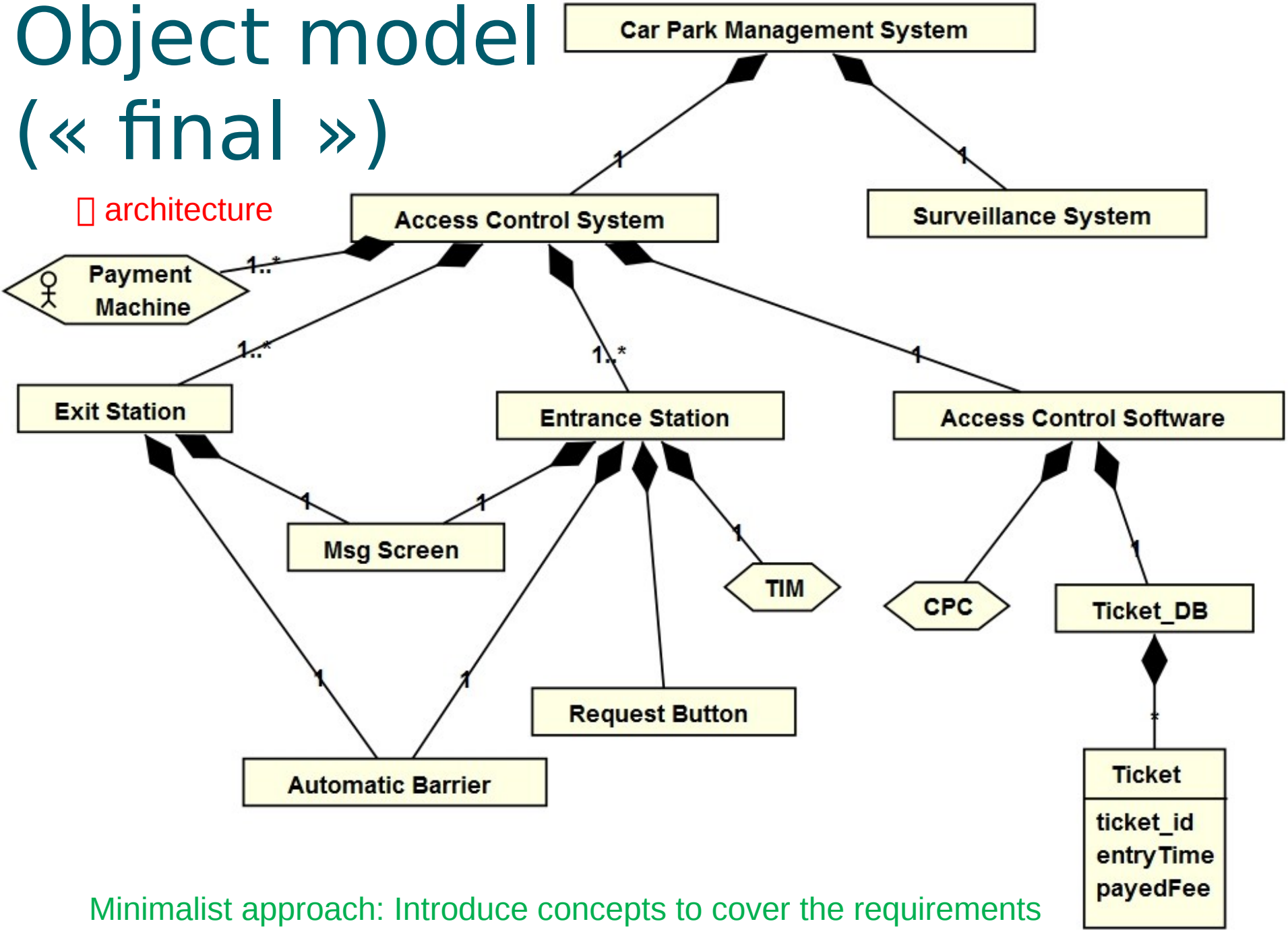
Object definition



- Agent Stereotype: if direct operational behaviour (not by means of an aggregation relationship)
- Attribute type: not needed at this stage
- Fill the model in operationalise the requirements

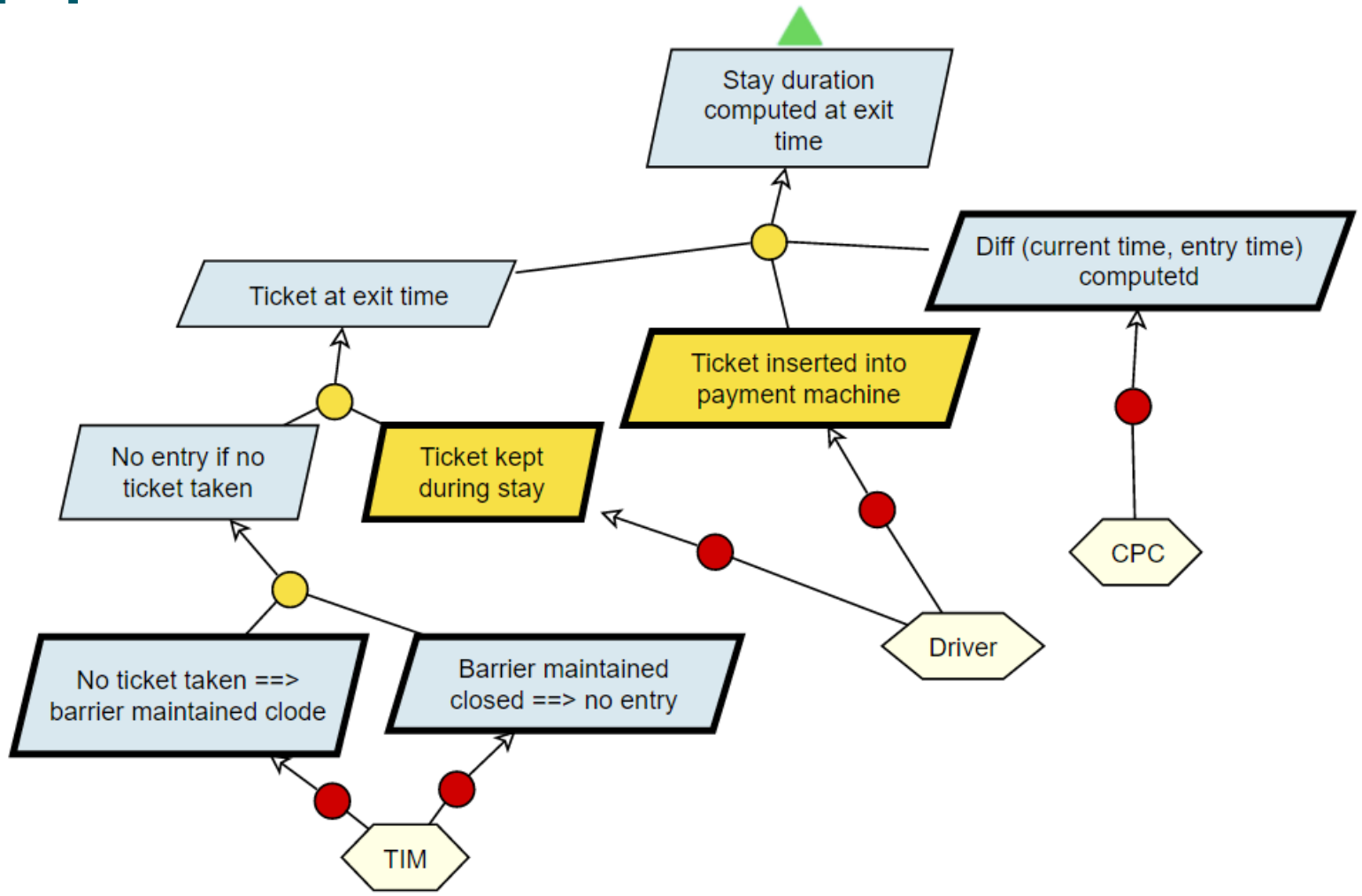
Object model (« final »)

 architecture



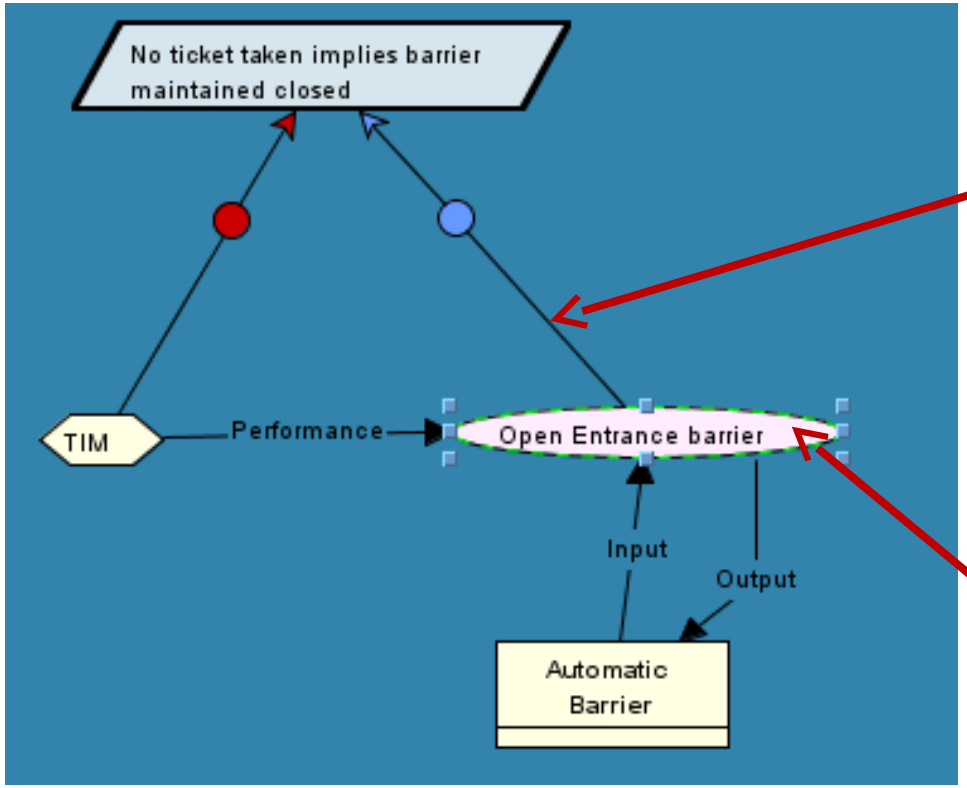
Minimalist approach: Introduce concepts to cover the requirements

Operationalisation I



- Concerned agents: CPC and TIM

Operationalisation : TIM



to Open Entrance barrier [Operationaliza...

Properties	Neighborhood	Documents

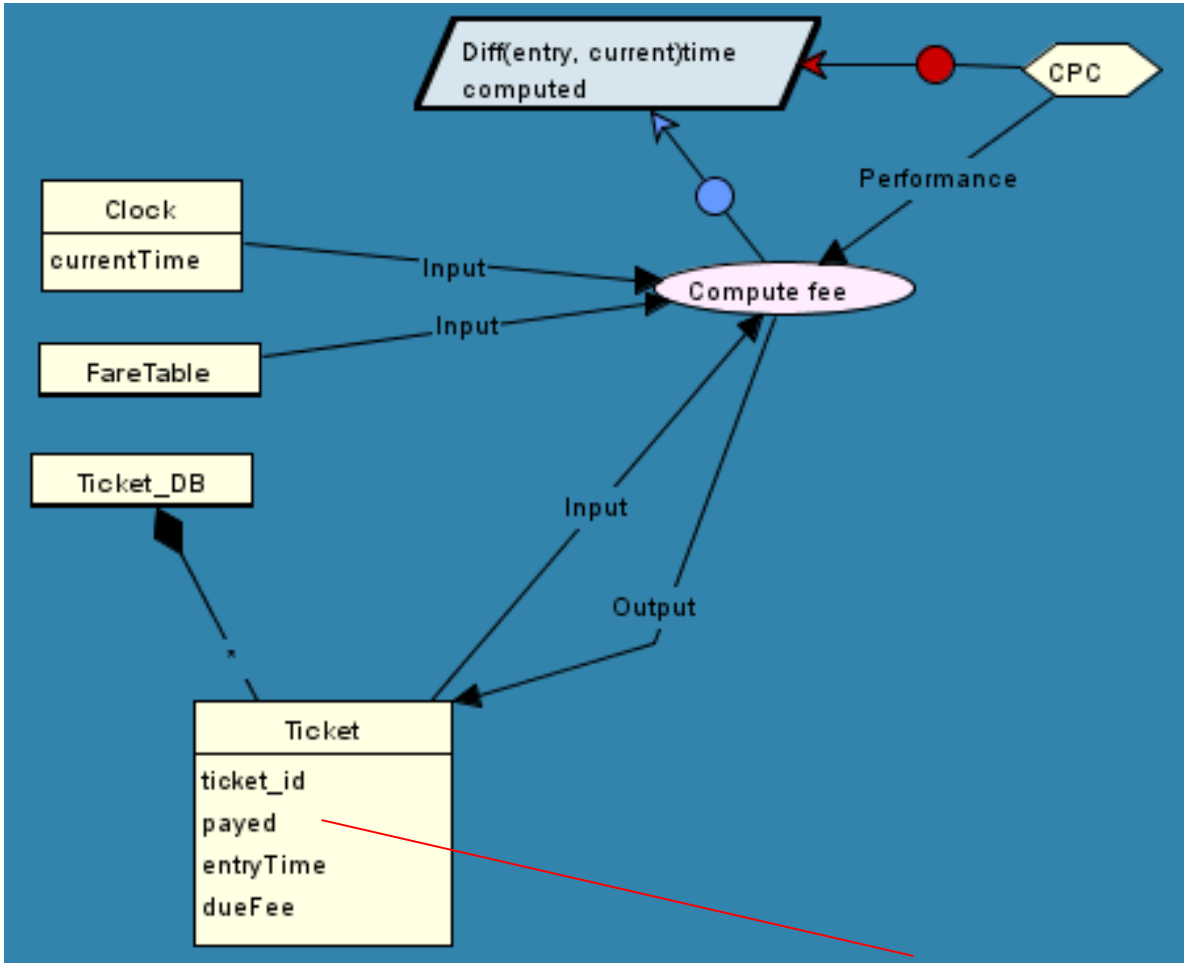
ReqPre	TicketTaken
ReqTrig	
FormalReqTrig	
ReqPost	
FormalReqPre	
FormalReqPost	

Open Entrance barrier [Operation]

Properties	Neighborhood	Documents

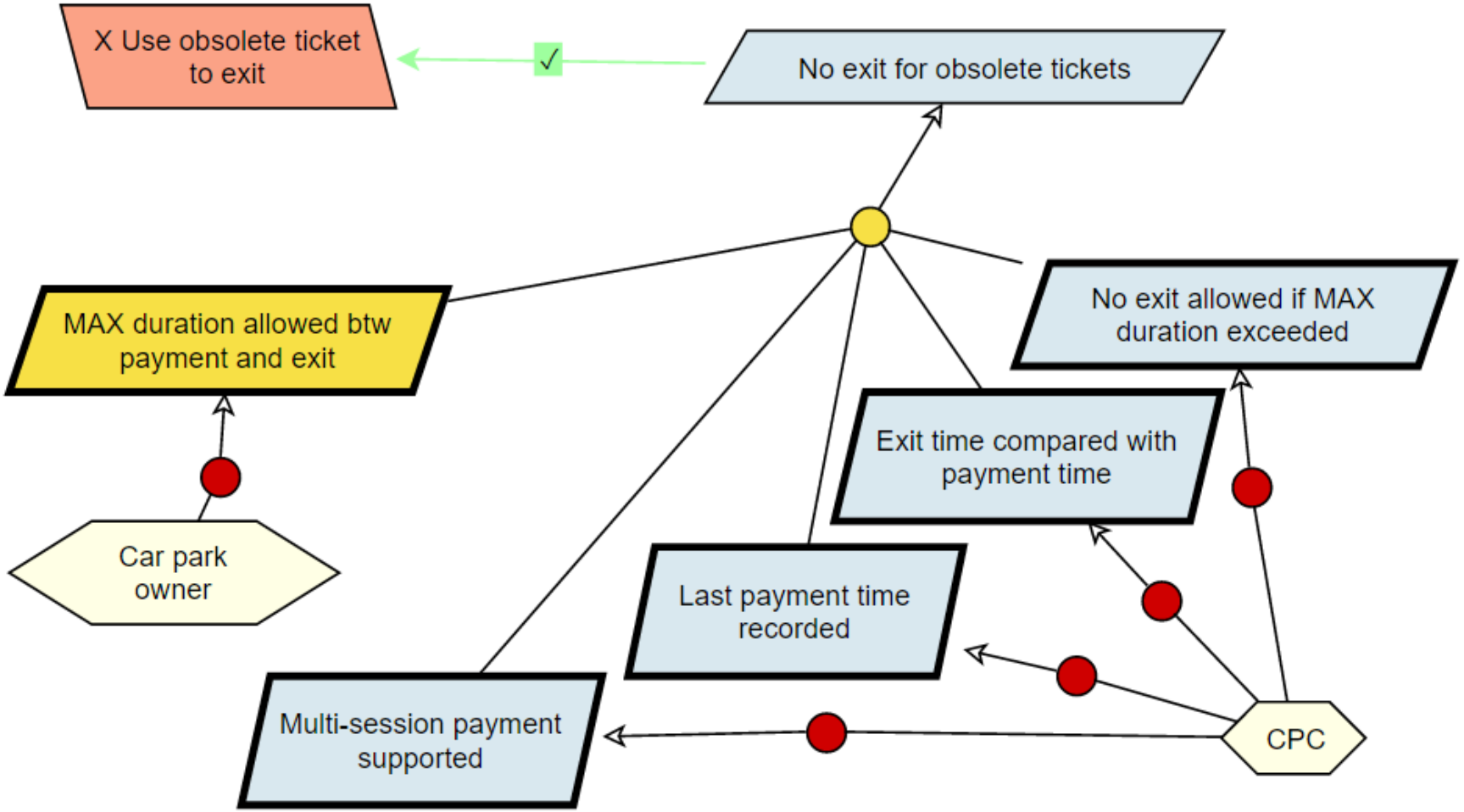
Name	Open Entrance barrier
Def	
DomPre	barrier is closed
DomPost	barrier is opened
FormalDomPre	
FormalDomPost	
Issue	

Operationalisation: CPC

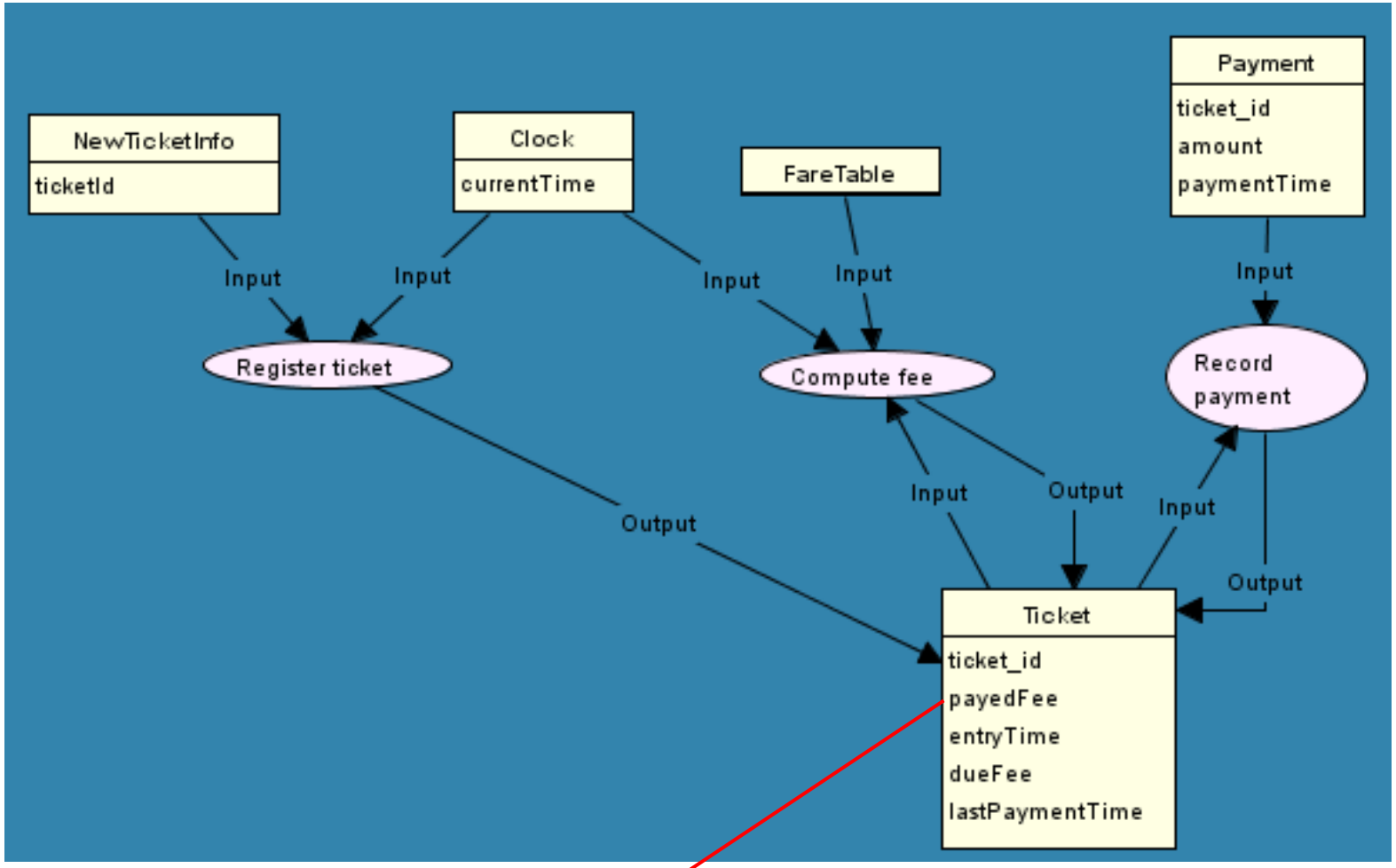


No multi-session yet!

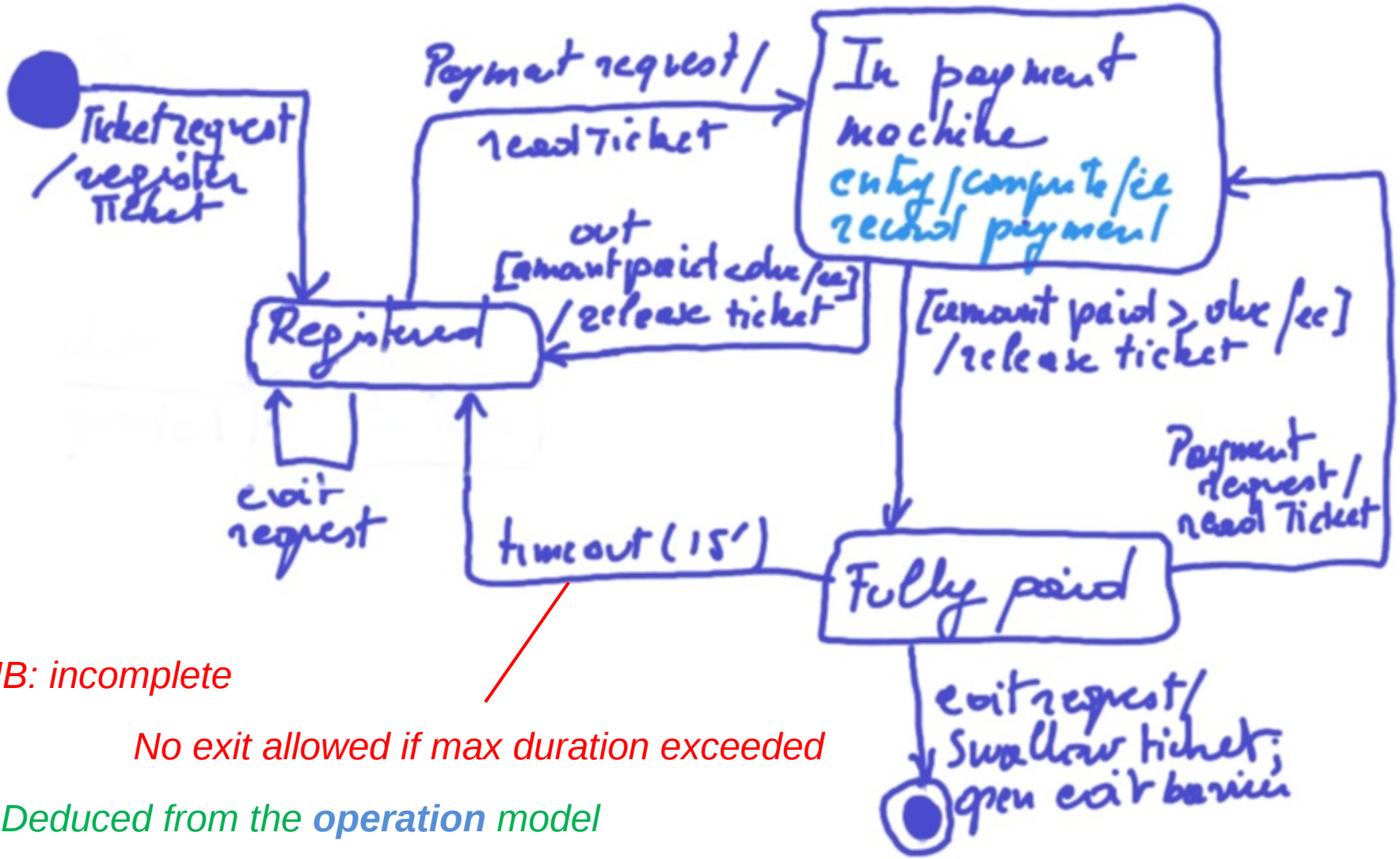
Operationalisation II



Multi-session payment



State-transition of a ticket



NB: incomplete

No exit allowed if max duration exceeded

*Deduced from the **operation** model*

Co Objectiver sion

GORE = keystone

- GORE:
  - Favors **creativity**
    - **alternatives** on AND/OR
  - Favors **completeness**:
    - Goals not refined, No responsible agents
    - Refinement tactics and patterns (missing goals)
    - Challenging goals (obstacles, threats)
  - Favors **communication** with the customers