Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN-Fakultät
Fachbereich Informatik
Arbeitsbereich SAV/BV (KOGS)

Image Processing 1 (IP1)
Bildverarbeitung 1

Lecture 23: High Level Vision

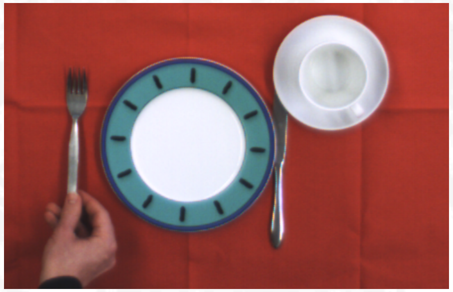
Winter Semester 2014/15

Dr. Benjamin Seppke
Prof. Siegfried Stiehl

High-level Vision

What are the tasks (is the scope) of high-level vision?

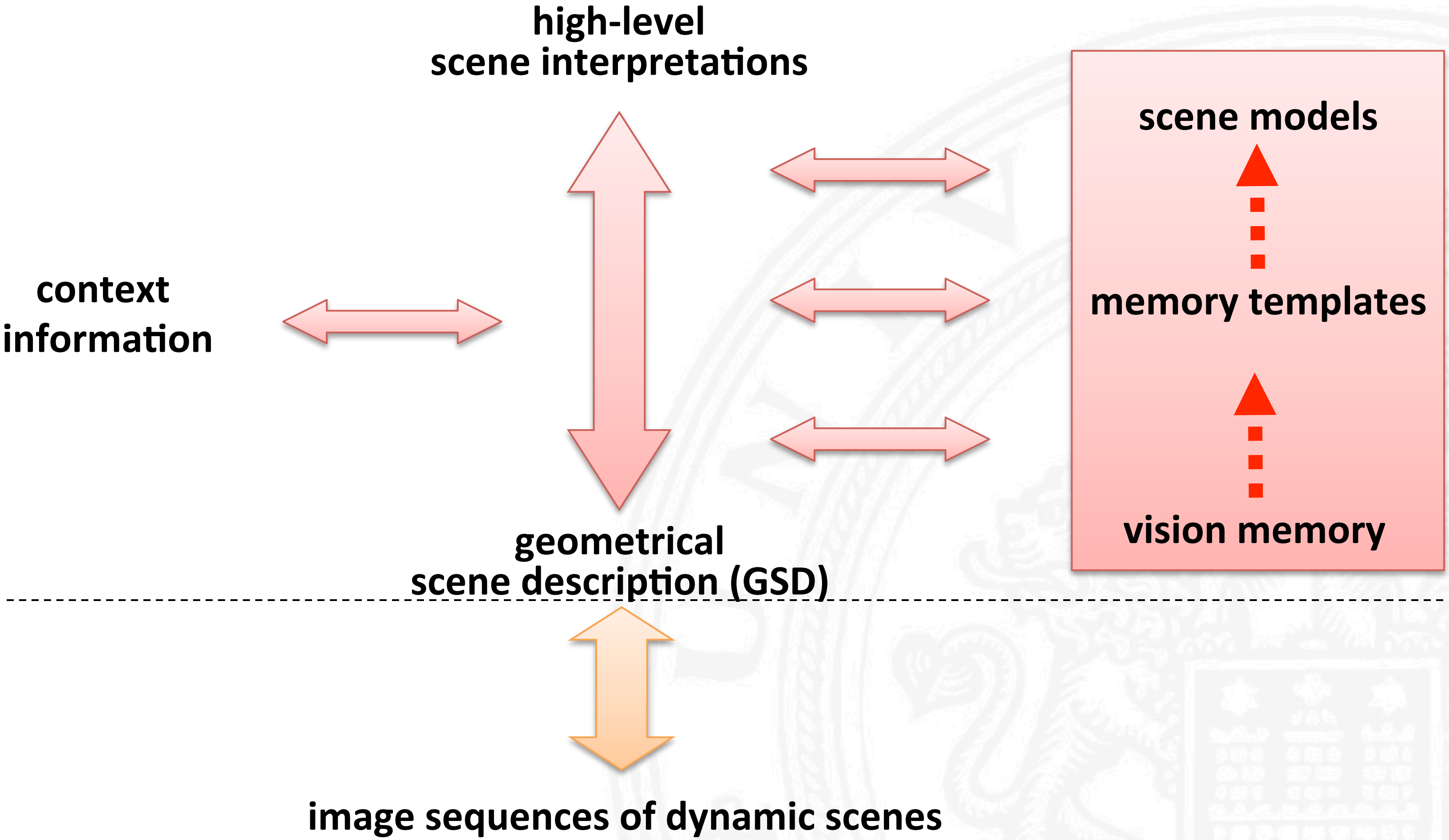
- Vision as silent-movie understanding
  - connecting to common-sense knowledge
  - understanding goal-oriented behaviour
  - vision in context
- Vision and acting
  - robot vision
  - goal-oriented vision, attention control
  - spatial and temporal reasoning
- Vision and learning
  - discovering reoccurring patterns
  - building models
  - predicting from experience



Topics of High-Level Vision

- Representing and recognizing structures consisting of several spatially and temporally related components (e.g. object configurations, situations, occurrences, episodes)
- Exploiting high-level knowledge and reasoning for scene prediction
- Understanding purposeful behaviour (e.g. obstacle avoidance, grasping and moving objects, behaviour in street traffic)
- Mapping between quantitative and qualitative descriptions
- Natural-language communication about scenes
- Learning high-level concepts from experience
- Connecting uncertain knowledge with logic-based reasoning

Basic Building Blocks for High-level Scene Interpretation



high-level scene interpretations

scene models

memory templates

vision memory

context information

geometrical scene description (GSD)

image sequences of dynamic scenes

Basic Representational Units

scene	spatially and temporally coherent real-world section
geometrical scene description (GSD)	scene description in terms of object locations in an image sequence
scene interpretation	scene description in terms of instantiated scene models (object configurations, occurrences, episodes, purposive actions)
memory record	memorized scene interpretation incl. imagery
memory template	generalized substructure of memory records
scene model	conceptual unit for scene interpretation

Towards Generic Models for Scene Interpretation

- Need for model-based approach
  - spatially and temporally coherent configurations
  - organising relevant knowledge
- Logic-based and probabilistic knowledge*
  - deduction, rules,
  - uncertainty, consistency
- Interface to low-level vision
  - signal-symbol interface
  - quantitative-qualitative mapping
- Interpretation strategies
  - bottom-up vs. top-down
  - varying context
  - prediction

*) Probabilistic issues will be treated later

Conceptual Units for Scene Interpretation

What kind of concepts must be represented for scene interpretation?

Concepts for

- object constellations
  e.g. laid-table, kitchen, parking ground, town
- activities, events, episodes
  e.g. operating a CD-player, one car overtaking another, playing soccer

Typical scene interpretation concepts describe entities composed of sub-entities related to each other in space and time. We call such entities "aggregates".

Note: The term "aggregate" will at times be used for an aggregate model (a conceptual description of a kind of aggregates) and at other times for an aggregate instance (a concrete occurrence of an aggregate). Hopefully, the context clarifies the intended meaning.

Aggregate Structure

Basic structure of a frame-based representation of an aggregate concept:

aggregate name
parent concepts
external properties
parts
constraints between parts

- *aggregate name* contains a symbolic ID
- *parent concepts* contains IDs of taxonomical parents
- *external properties* provide a description of the aggregate as a whole
- *parts* describe the subunits out of which an aggregate is composed
- *constraints* specify which relations must hold between the parts

Occurrence Models

Basic ingredients:

- relational structure
- taxonomy
- partonomy
- spatial relational language
- temporal relational language
- object appearance models

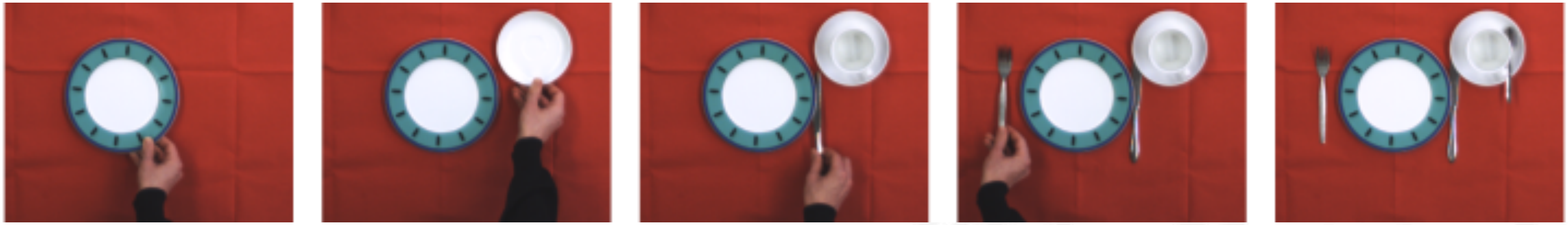
- An occurrence model describes a class of occurrences by:
  - properties
  - sub-occurrences (= components of the occurrence)
  - relations between sub-occurrences
- A primitive occurrence model consists of
  - properties
  - a qualitative predicate
- Each occurrence has a begin and end time point

Occurrence Model for Overtaking

name:	**overtake :local-name ov**
parents:	**:is-a occurrence-model**
arguments:	**(?veh1 :is-a vehicle)**
	(?veh2 :is-a vehicle)
properties:	**(ov.B ov.E)**
parts :	**(mv1 :is-a (move ?veh1 mv1.B mv1.E))**
	(mv2 :is-a (move ?veh2 mv2.B mv2.E))
	(bh :is-a (behind ?veh1 ?veh2 bh.B bh.E))
	(bs :is-a (beside ?veh1 ?veh2 bs.B bs.E))
	(bf :is-a (before ?veh1 ?veh2 bf.B bf.E))
	(ap :is-a (approach ?veh1 ?veh2 ap.B ap.E))
	(rc :is-a (recede ?veh1 ?veh2 rc.B rc.E))
constraints:	**(ov.B = bh.B)**
	(ov.E = bf.E)
	(ap :during mv1)
	(ap :during mv2)
	(rc :during mv1)
	(rc :during mv2)
	(bh :overlaps bs)
	(bs :overlaps bf)
	(bh :during ap)
	(bf :during rc)

Aggregate format may vary according to expressiveness of knowledge representation language and syntactic conventions

Table-laying Scenario



Important high-level characteristics:

- correlated multiple object motion
- intended actions
- influence of context (temporal, spatial, task context)
- qualitative spatial and temporal relations
- uncertainty
- smart room learning context (supervised, unsupervised)
- interface with common sense

Table-laying scenario of project CogVis:

Stationary cameras observe living room scene and recognize meaningful occurrences, e.g. placing a cover onto the table.

Occurrence Model for Placing a Cover

```
name:          place-cover
parents:       :is-a agent-activity
parts:         pc-tp1 :is-a (transport with (tp-obj :is plate))   %transport of a plate
               pc-tp2:is-a (transport with (tp-obj :is saucer))   %transport of a saucer
               pc-tp3 :is-a (transport with (tp-obj :is cup))     %transport of a cup
               pc-cv :is-a cover                                          %cover configuration
properties:    tb, te :is-a timepoint           %begin and end timepoint of place-cover
constraints:   pc-tp1.tp-ob = pc-cv.cv-pl       %transport-plate object same as cover-plate
               pc-tp2.tp-ob = pc-cv.cv-sc       %transport-saucer object same as cover-saucer
               pc-tp3.tp-ob = pc-cv.cv-cp       %transport-cup object same as cover-cup
               pc-cv.tb ≥ pc-tp1.te             %cover begins after plate transport
               pc-cv.tb ≥ pc-tp2.te             %cover begins after saucer transport
               pc-cv.tb ≥ pc-tp3.te             %cover begins after cup transport
               pc-tp3.tp-te ≥ pc-tp2.tp-te      %cup transport ends after saucer transport
               tb = pc-tp1.tb min pc-tp2.tb min pc-tp3.tb         %begin of place-cover
               te = pc-tp1.te max pc-tp2.te max pc-tp3.te         %end of place-cover
               te ≤ tb + 80Dt                   %place-cover may not last more than 80 time units
```

Model for a Cover Configuration

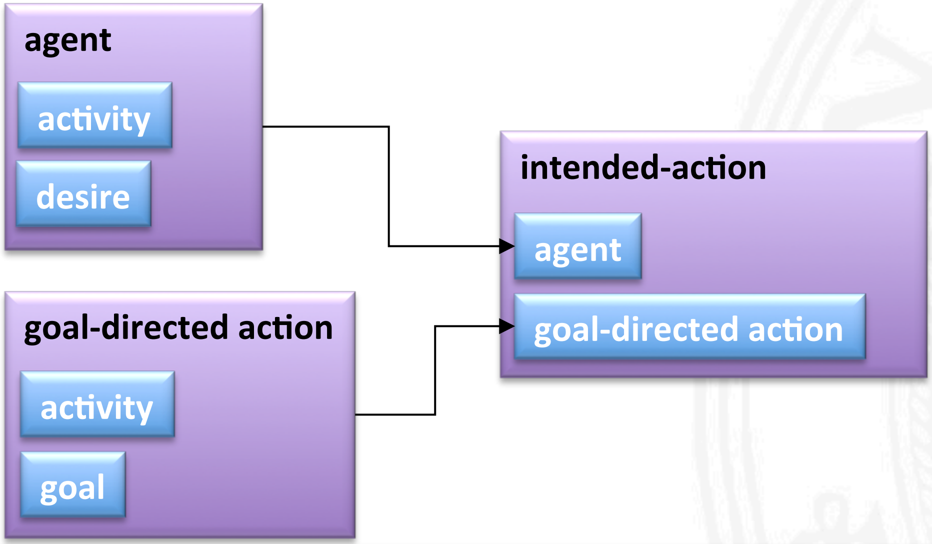
```
name:           cover
parents:        :is-a configuration
parts:          cv-pl :is-a plate
                cv-sc :is-a saucer
                cv-cp :is-a cup
                cv-tt :is-a table-top
properties:     w, h, tb, te                  %width and height of cover
constraints:    cv-sc.pos NE cv-pl.pos        %saucer position northeast of
                                                plate position
                cv-sc.rim CLOSE cv-pl.rim     %saucer rim close to plate rim
                cv-cp.pos = cv-sc.pos
                cv-tt.rim SO cv-pl.rim        %table-top rim south of plate rim
```

Spatial relations NO (north), NE (northeast), ... , SO (south), ... , CLOSE must be defined and computable based on parts properties.

Models for Intention Recognition

Intended actions may be described by aggregates which connect observable actions with (unobservable) intentions of an actor.

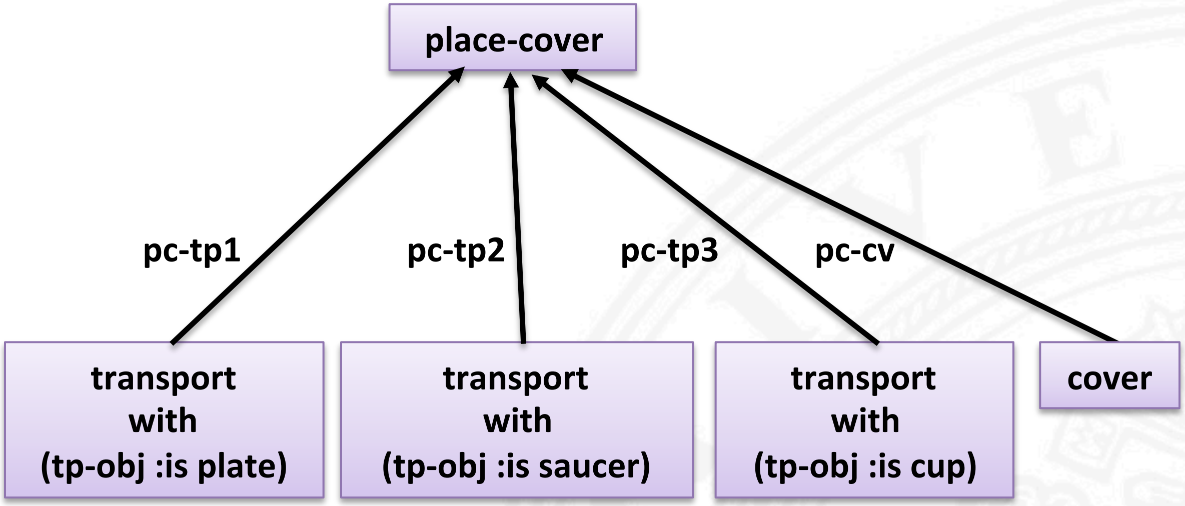
name:	**intended-place-cover**
parents:	**:is-a intended-action**
parts:	**ipc-pc :is-a place-cover**
	ipc-ag :is-a agent with (ipc-ag.desire = ipc-pc.goal)
properties:	**tb, te :is-a timepoint**
constraints:	**(temporal, spatial and other constraints on parts)**



If an action is known to be goal-directed and an agent performs such an action, the agent is ascribed the intention to attain the goal.

Parts Structure

Inferential structure between aggregates and their parts


place-cover
pc-tp1
pc-tp2
pc-tp3
pc-cv
transport with (tp-obj :is plate)
transport with (tp-obj :is saucer)
transport with (tp-obj :is cup)
cover

"In a place-cover occurrence one will see transport occurrences with plate, saucer and cup, and a cover configuration."

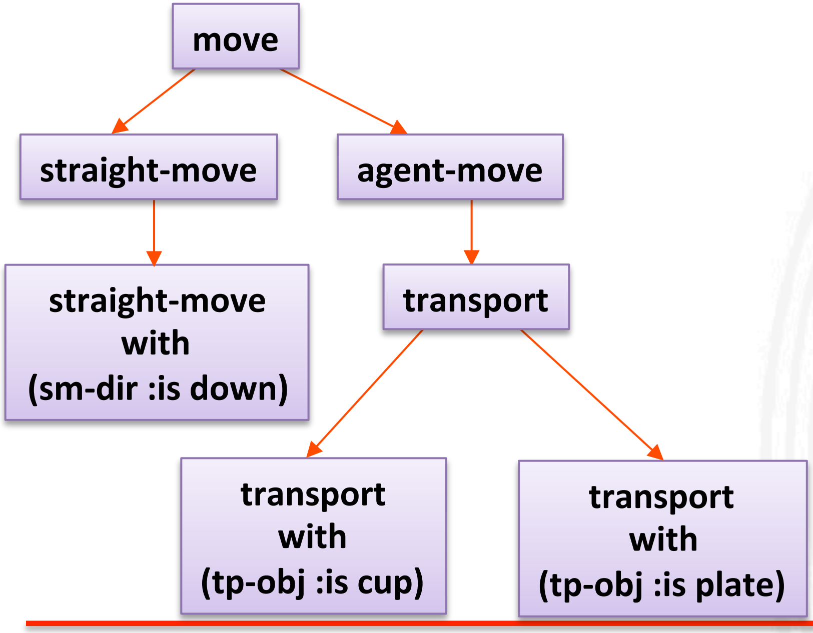
Note that redundant parts could be added, e.g. plate, saucer, cup, table-top and linked to other parts by equality constraints. Redundant parts may be useful for triggering part-whole reasoning ("*If you see a plate and the transport of a saucer, hypothesise a place-cover*").

Forming a Taxonomical Hierarchy

Remember:

- A concept denotes a set of "objects".
- "Objects" may be physical objects, occurrences, configurations, ...
- A specialisation denotes a subset of a parent concept.
- Different kinds of "objects" require different hierarchies.

motion of a physical object



move

straight-move

agent-move

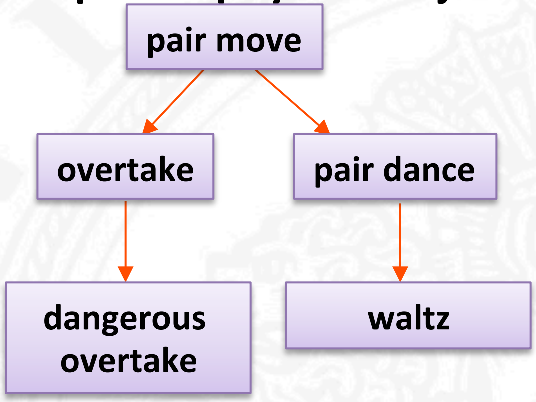
straight-move with (sm-dir :is down)

transport

transport with (tp-obj :is cup)

transport with (tp-obj :is plate)

motion of a pair of physical objects



pair move

overtake

pair dance

dangerous overtake

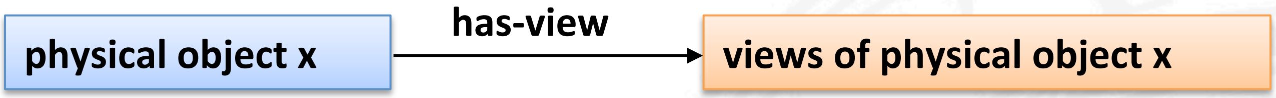
waltz

Note that *pair move* could also be represented as a specialisation of *move* (a moving object specialised by a relation to an accompanying object).

Physical Objects and Views

Representations of physical (3D) objects must be distinguished from representations of evidence obtained by sensors, e.g. 2D views.

Suggested conceptual representation:



physical object x

has-view

views of physical object x

In a conceptual knowledge base ...

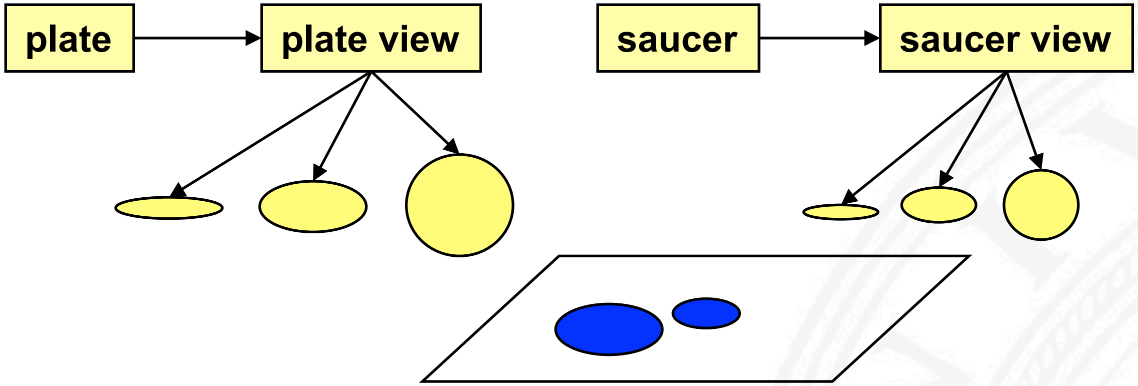
- a physical object model describes properties of 3D objects irrespective of sensors,
- a view model describes the responses of a specific sensor for a 3D object.

Views may alternatively be represented as "properties" of physical objects, but the explicit representation above emphasises the dependency on sensors and alleviates multi-sensor modelling.

Results of Low-level Image Analysis

Assumptions

- **Low-level image analysis provides evidence which can be matched with *object views* of the conceptual knowledge base.**



view descriptions of conceptual knowledge base

evidence of low-level image analysis

- **Evidence is represented in metric space.**
- **Evidence may be**
  - **regions corresponding to objects**
  - **blobs corresponding to object parts**
  - **descriptive features around interest points**

depending on sophistication of object recognition and categorisation

Tasks of Signal-Symbol Interface

- Matching evidence with views
  - bottom-up: classification
  - top-down: hypothesis verfication
- Depth management
  - maintaining a qualitative depth map
  - maintaining consistency of occlusion hypotheses
- Computing predicates on perceptual primitives
  - providing useful primitives for inter-object relations
  - enabling temporal segmentation

All of these tasks are still research topics.

Some ideas and possible approaches will be shown in the following slides.

Matching Evidence with Views

Bottom-up classification

- *Assign evidence to one of several view classes.*
- Model-based recognition problem with view classes as models.
- In a probabilistic setting same as Bayesian classification, except that a priori class probabilities depend on interpretation context.

Top-down hypothesis verfication

- *Check compatibility of top-down view hypothesis with available evidence and other top-down hypotheses.*
- Checking with evidence is similar to bottom-up classification, except that model is given and evidence is selected.
- Checking with other top-down hypotheses is a harder task, as all hypotheses may have uncertainty ranges. How can several hypotheses with uncertain views and locations fit into an image, observing factual evidence and occlusion rules?

Predicates on Perceptual Primitives

Useful for describing relations between objects (e.g. "close-to", "beside", "parallel") and inducing primitive occurrences (e.g. "approach", "turn")

1. Measurements of perceptual primitives

- Evidence objects provide reference features:
  - locations (center of gravity, corners, surface markings, etc.)
  - lines (edges, surface markings, axes of minimal inertia, etc.)
  - orientations (inate, motion-dependent, viewer-dependent)
  - size, shape, photometric properties
- Measurements between geometric reference features:
  - distance, relative orientation, orientation of location difference vector
  - temporal derivatives thereof

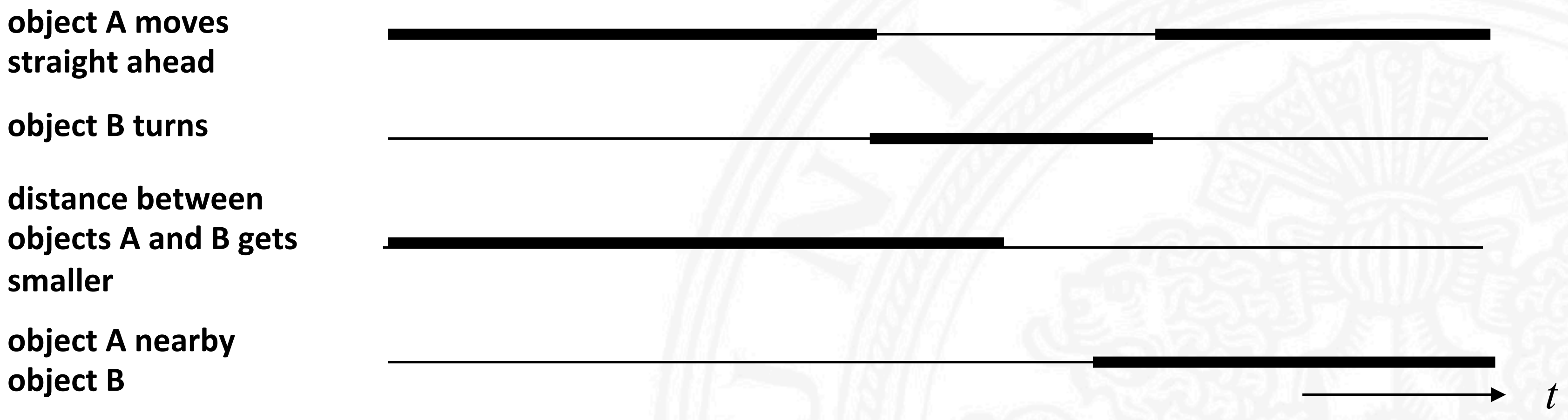
2. Qualitative predicates

- Qualitatively constant values e.g. constant orientation, constant distance
- Values within a certain range e.g. topological relations, degrees of nearness, typical speeds, slowing down, inceasing distance

Primitive Occurrences

A primitive occurrence is a symbolic entity involving one or more evidence objects for which a qualitative predicate is true over a time interval.

Primitive occurrences provide the raw material for the interpretation of time-varying scenes.



object A moves straight ahead

object B turns

distance between objects A and B gets smaller

object A nearby object B

In a natural scene, one may observe many time-dependent perceptual primitives and determine many primitive occurrences. Hence it may be useful to compute primitive occurrences on demand (attention driven).

Primitive Occurrences in Traffic Scenes

B. Neumann: Natural Language Description of Time-Varying Scenes. In: Semantic Structures, D. Waltz (Hrsg.), Lawrence Erlbaum, 167-206, 1989

exist
move
decelerate, accelerate
turn_left, turn_right
increasing_distance, reducing_distance
along, across
in_front_of, behind, beside
on, above, under, below
at, near_to
between
(and others)

Note that qualitative predicates are often (but must not be) part of natural language.

For technical applications one may use technical (artificial) qualitative predicates, e.g.

v50 (= 45 ≤ v ≤ 55 km/h)

shape_x (= shape_index ≤ 4.174)

Temporal vs. Spatial Decomposition of Scenes

Temporal decomposition

- **by temporal segmentation:**
  constancies of time-dependent properties of an image sequence
- **by model matching:**
  occurrences which obey a model

Compare with spatial decomposition

- **by spatial segmentation:**
  image regions with spatially constant (uniform) properties
- **by model matching:**
  image regions which obey a model

Stepwise Construction of Scene Interpretations

Given taxonomical and compositional concept hierarchies, there are five kinds of interpretation steps for constructing interpretations consistent with evidence:

Evidence matching
Assignment of evidence to object view classes or verification of view hypotheses.

Aggregate instantiation
Inferring an aggregate from (not necessarily all) parts

Instance specialization
Refinements along specialization hierarchy or in terms of aggregate parts

Instance expansion
Instantiating parts of an instantiated aggregate

Instance merging
Merging identical instances constructed by different interpretation steps

Repertoire of interpretation steps allows flexible interpretation strategies
e.g. mixed bottom-up and top-down, context-dependent, task-oriented

Basic Interpretation Algorithm

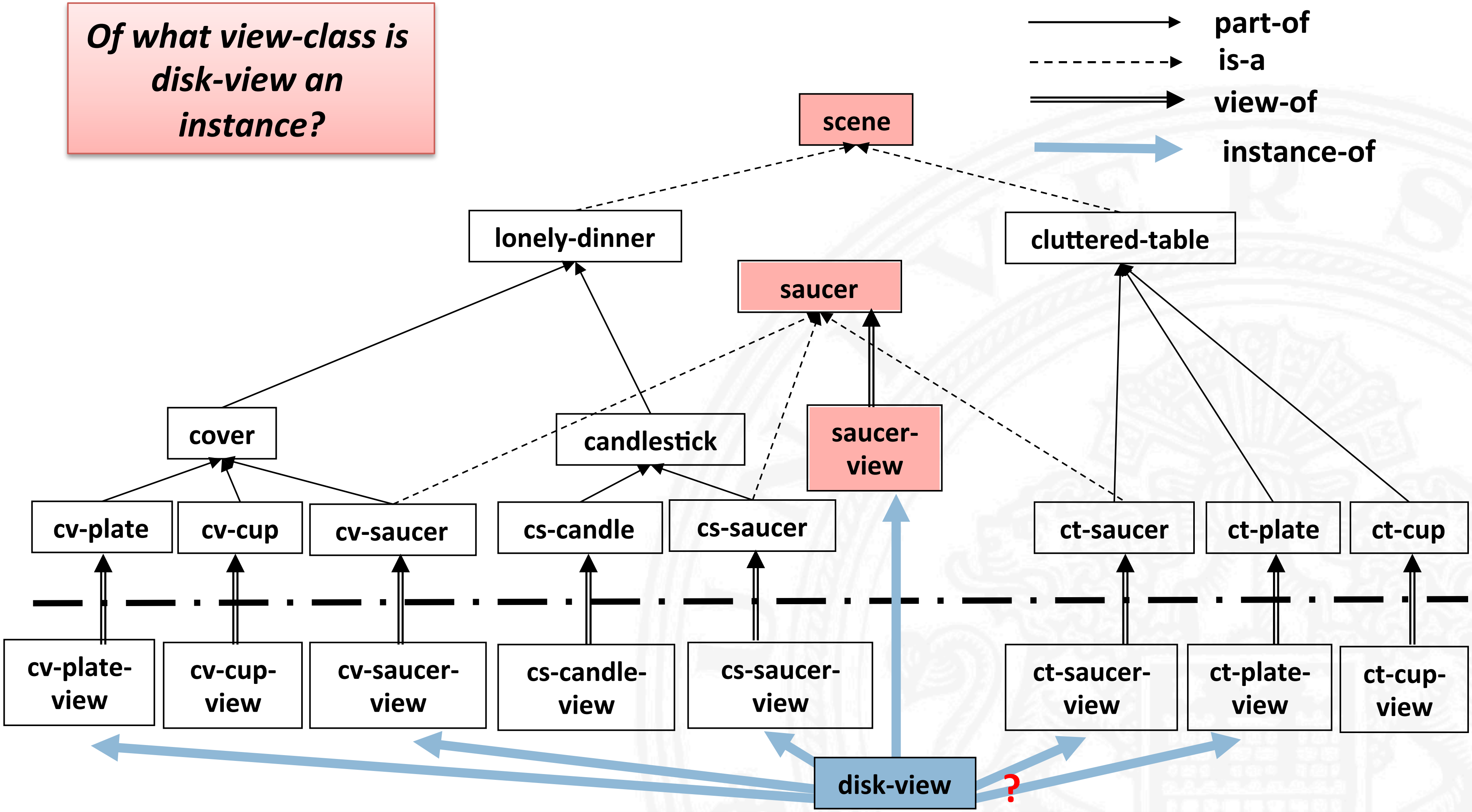
```
Enter context information
Repeat
        Check for goal completion
        Check for new evidence
        Determine possible interpretation steps and update agenda
        Select from agenda one of
                {  evidence matching,
                   aggregate instantiation,
                   aggregate expansion,
                   instance specialization,
                   parameterization,
                   constraint propagation }
        Check for conflict
end
```

Conflict = unsatisfiable constraint net

→need for backtracking or parallel alternative threads

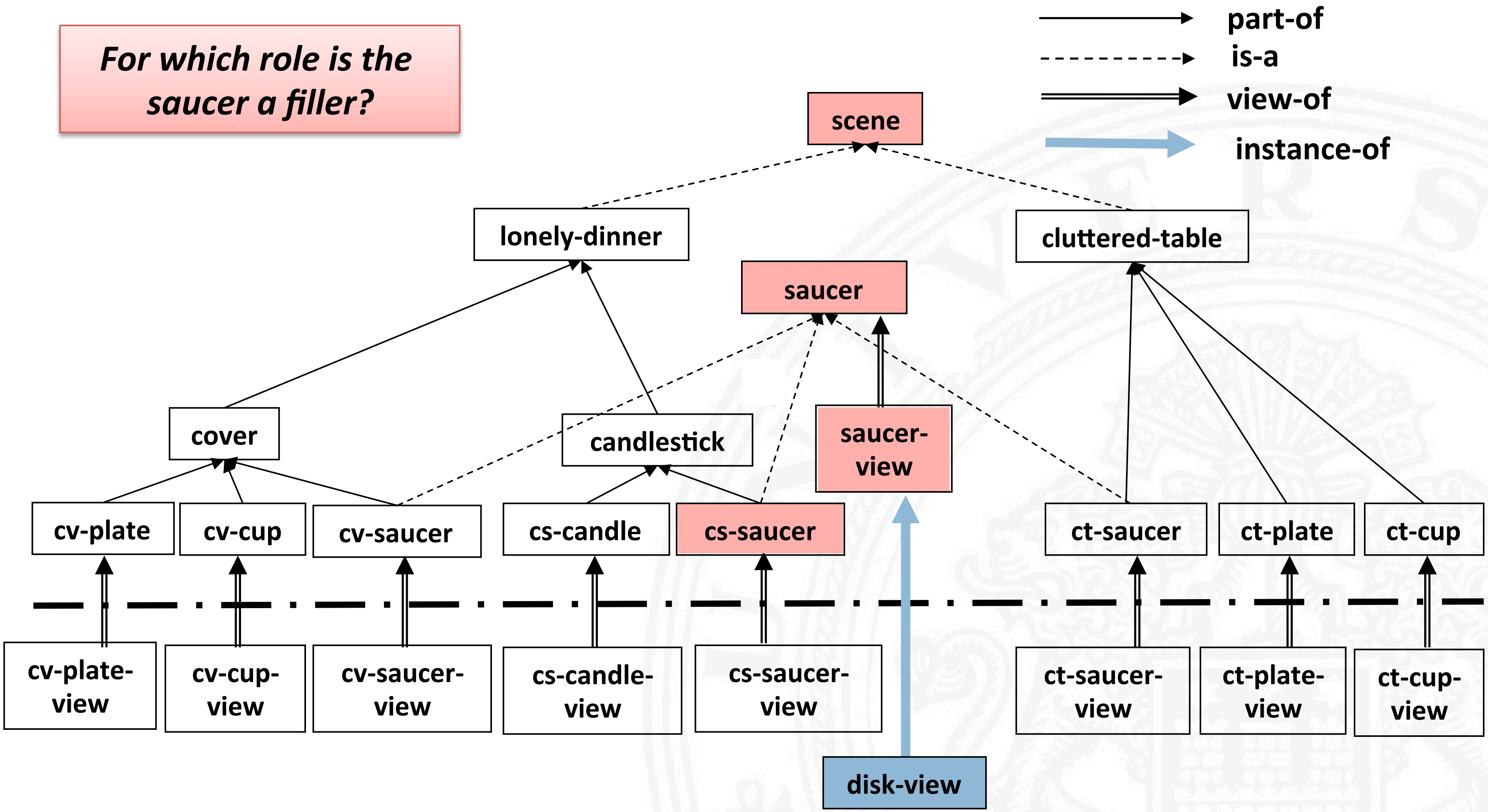
Example for Interpretation Steps I

Of what view-class is disk-view an instance?



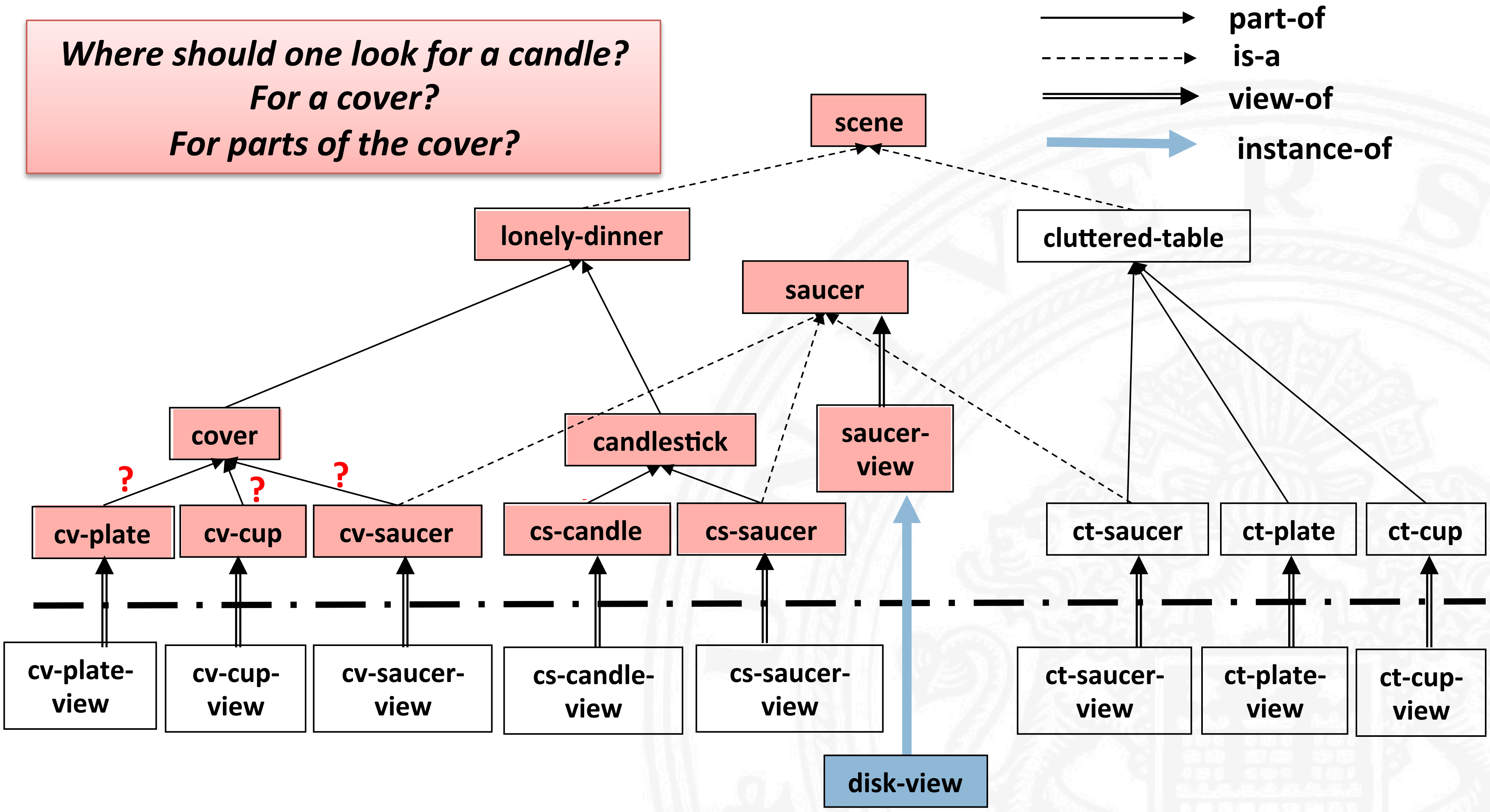
Example for Interpretation Steps II

For which role is the saucer a filler?

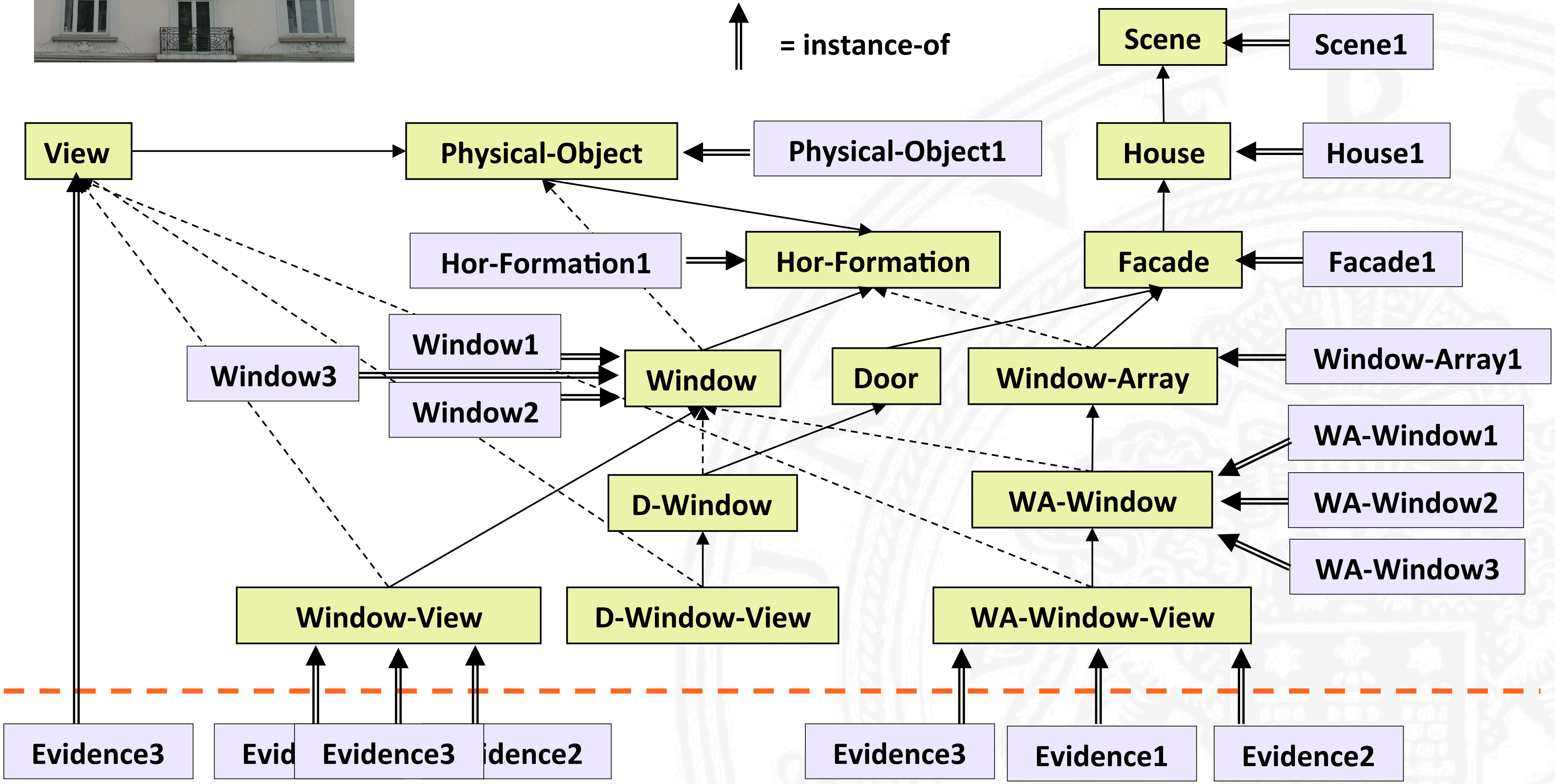


Example for Interpretation Steps III

Where should one look for a candle?
For a cover?
For parts of the cover?



Facade Interpretation



Hallucination Space

Interpretation steps allow to liberally hypothesise ("hallucinate") parts of aggregates and to come up with multiple alternative interpretations.

The validity of an interpretation depends on the available evidence and the readiness to believe in an interpretation based on scarce or no evidence.

Hallucination is desirable

- to predict future occurrences,
- to cope with occluded or unobserved evidence.

Hallucination is problematic because

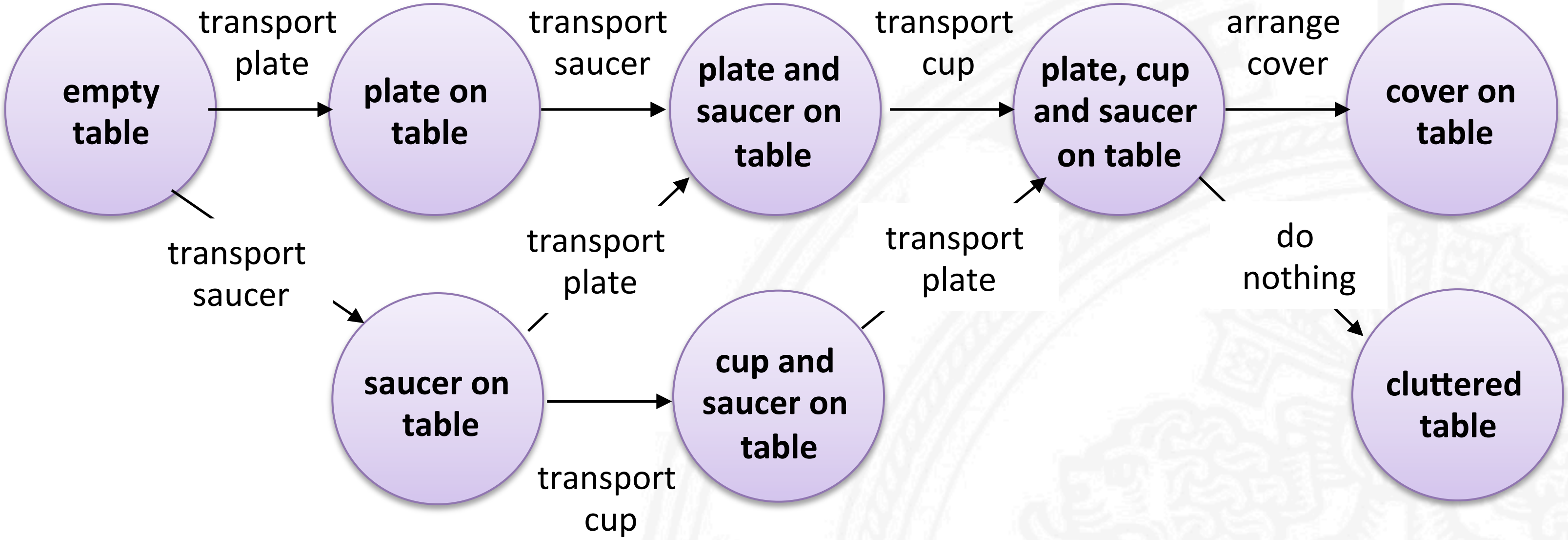
- many alternative interpretations are permitted,
- a single interpretation may include many unsupported hypotheses.

Practical use of hallucination for scene interpretation requires that interpretation steps are guided by a preference measure (later in this course).

State Transition Models (1)

Sometimes occurrences can be descibed as transitions between states.

Placing a cover:

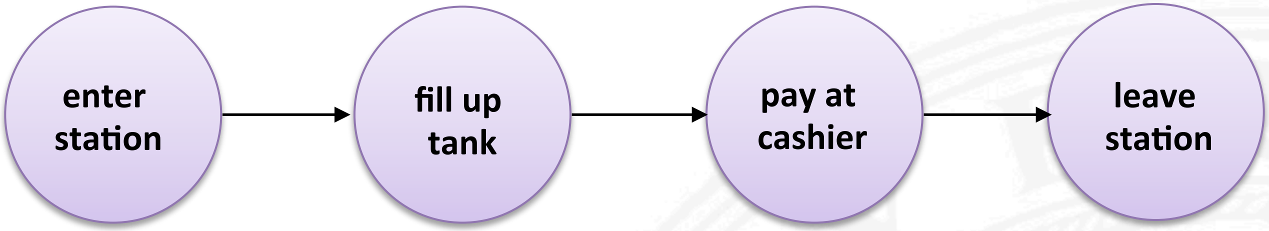


State transition models provide an explicit representation of

- states = intervals with specific constant properties
- state transitions = events leading from one state to another
- partial temporal ordering of states based on temporal succession

State Transition Models (2)

States need not be stationary, e.g. filling up at a gas station:



Here the transitions denote "temporal succession" without specifying events or activities associated with the transitions.

State transition models are atractive because they allow to abstract from many details and also relate to probabilistic Markov Models.

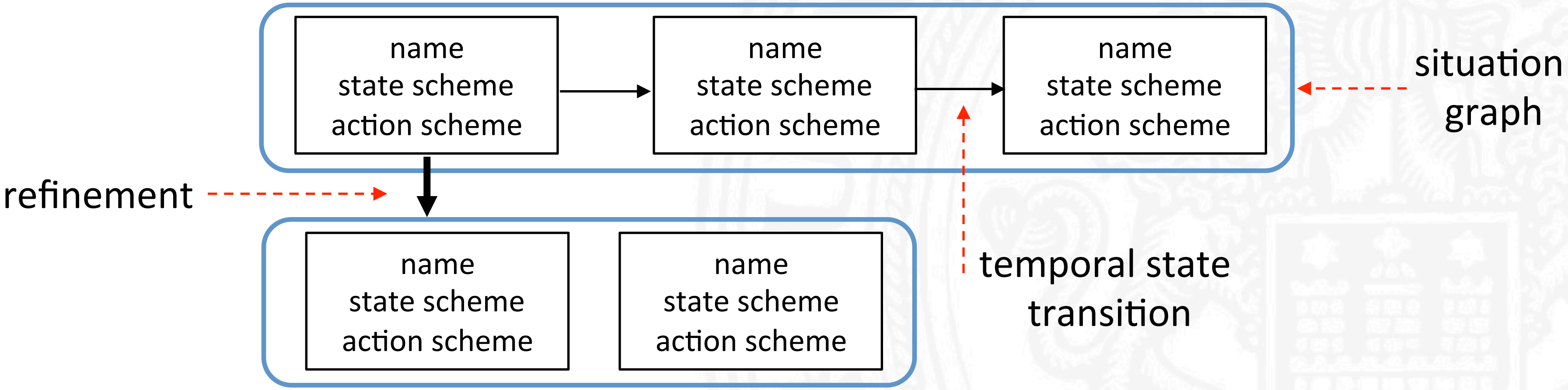
But the operational semantics are not always clear, e.g.

- Are there temporal constraints for state transitions?
- If a state is defined by several predicates -
- what is the temporal extent of a state?
- What is a generalisation or a specialisation of a state transition model?

Situation Graph Trees (SGTs)

H.-H. Nagel, Natural Language Description of Image Sequences as a Form of Knowledge Representation, Proc. KI-99, Springer, 1999, pp. 45-60

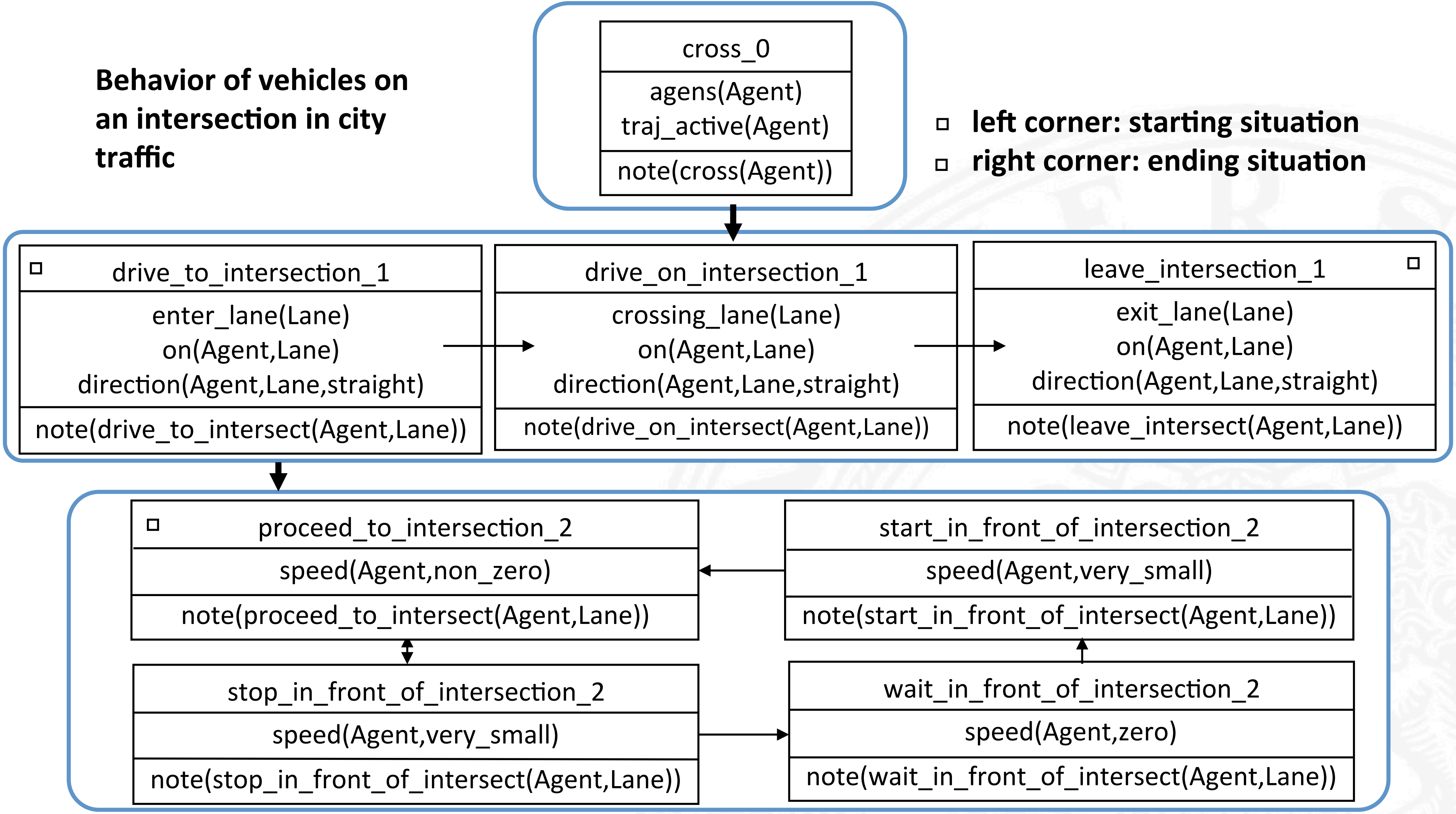
- Knowledge about agent behavior is expressed in terms of situations an agent can be in.
- A situation scheme is a generic situation description.
  It consists of a state scheme and an action scheme.
- If the predicates of the state scheme are satisfied, an agent instantiates the situation and is expected to execute the actions of the action scheme.
- Transitions between situations describe a temporal change.
- Situation schemes are refined in a tree structure.



Example of a Situation Graph Tree

Behavior of vehicles on an intersection in city traffic

- □ **left corner: starting situation**
- □ **right corner: ending situation**



cross_0
agens(Agent)
traj_active(Agent)
note(cross(Agent))

□ drive_to_intersection_1
enter_lane(Lane)
on(Agent,Lane)
direction(Agent,Lane,straight)
note(drive_to_intersect(Agent,Lane))

drive_on_intersection_1
crossing_lane(Lane)
on(Agent,Lane)
direction(Agent,Lane,straight)
note(drive_on_intersect(Agent,Lane))

leave_intersection_1 □
exit_lane(Lane)
on(Agent,Lane)
direction(Agent,Lane,straight)
note(leave_intersect(Agent,Lane))

□ proceed_to_intersection_2
speed(Agent,non_zero)
note(proceed_to_intersect(Agent,Lane))

start_in_front_of_intersection_2
speed(Agent,very_small)
note(start_in_front_of_intersect(Agent,Lane))

stop_in_front_of_intersection_2
speed(Agent,very_small)
note(stop_in_front_of_intersect(Agent,Lane))

wait_in_front_of_intersection_2
speed(Agent,zero)
note(wait_in_front_of_intersect(Agent,Lane))

Behavior Recognition with SGTs

(See details in Nagel 99)

Basic recognition algorithm is graph traversal:

```
Startnodes = {root node of SGT}.
VERIFY(startnodes)
Try to instantiate node of startnodes.
        A     Instantiated node is leaf node: Follow prediction arrows until
              situation graph is completely instantiated.
        B:    Instantiated node is not a leaf node: Obtain startnodes of
              refined situation graph and VERIFY(startnodes).
Return to next higher level of SGT.
```

Note correspondences:

- situation graph = aggregate
- situation scheme = part of aggregate
- SGT = specialisation hierarchy of aggregates

Scenarios

B. Georis, M. Mazière, F. Brémond, M. Thonnat: Evaluation and Knowledge Representation Formalisms to Improve Video Understanding. Proc. ICVS-06, 2006.

- **Scenario** symbolic description of a long-term activity

  e.g. "fighting", "vandalism"

  Scenarios may be structured into hierarchies (subscenarios, etc.)

- **Event** significant change of States

  "enters", "stands up", "leaves"

- **State** a spatio-temporal property involving one or several actors in a time interval

  e.g. "close", "walking", "seated"

Types of States and Events

- **Several types of States:**

– posture	e.g. lying, crouching, standing
– direction	e.g. towards the right, towards the left, leaving, arriving
– speed	e.g. stopped, walking, running
– distance/object	e.g. close, far
– distance/person	e.g. close, far
– posture/object	e.g. seated, any

- **Several types of Events:**

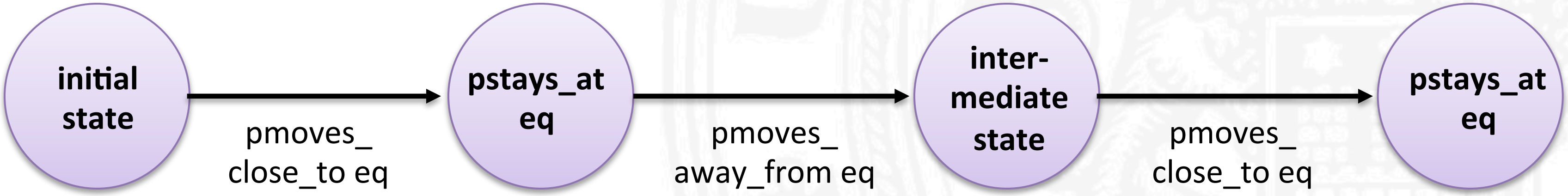
– person	e.g.	falls down, crouches down, stands up, goes right, goes left, goes away, arrives, stops, starts running
– person & zone		leaves, enters
– person & equipment		moves close to, sits on, moves away from
– 2 persons		moves close to, moves away from

Example Scenario "Vandalism"

Vandalism scenario description:

```
Scenario(vandalism_against_ticket_machine,
        Physical_objects((p : Person),
                                (eq : Equipment, Name= “Ticket_Machine”) )
        Components( (event s1: pmoves_close_to eq)
                                (state s2: pstays_at eq)
                                (event s3: pmoves_away_from eq)
                                (event s4: pmoves_close_to eq)
                                (state s5: pstays_at eq) )
        Constraints(   (s1 != s4) (s2 != s5)
                                (s1 before s2) (s2 before s3)
                                (s3 before s4) (s4 before s5) ) ) )
```

Notation as state transition graph:



initial state

pmoves_ close_to eq

pstays_at eq

pmoves_ away_from eq

inter-mediate state

pmoves_ close_to eq

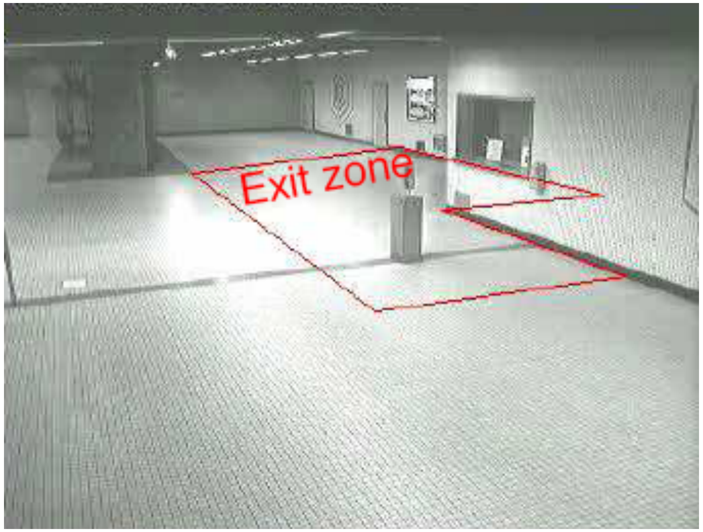
pstays_at eq

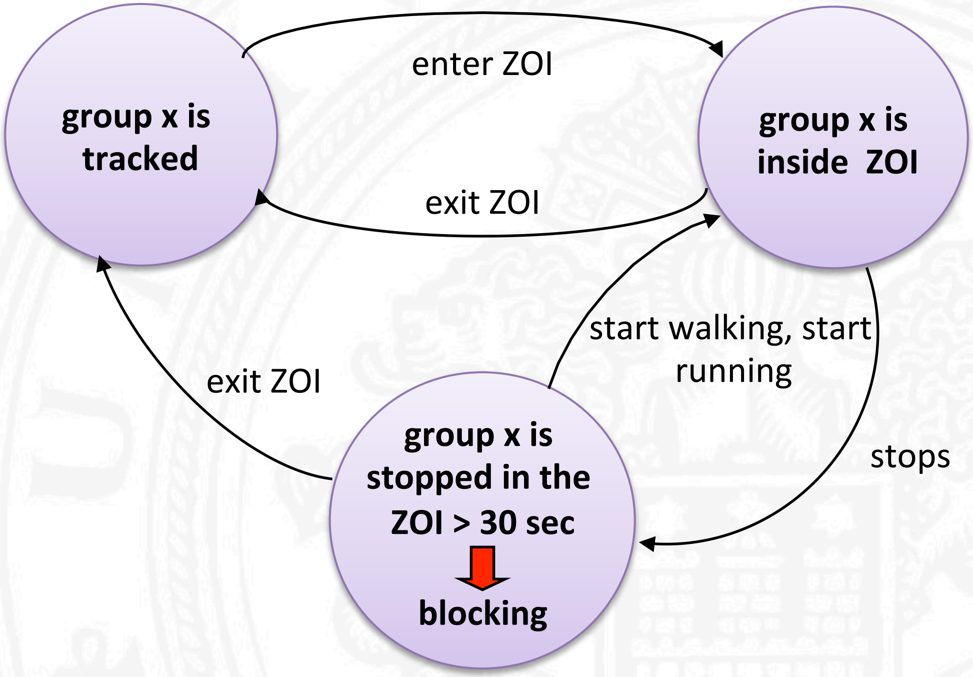
Scenario Recognition by State Transitions

- States and Events: Recognition by specific routines and classification
- Scenarios: Recognition based on finite state automata and propagation of temporal constraints

Example: Finite state automaton for scenario "A group of people blocks an exit" in a subway station monitoring task

Zone of interest (ZOI)




group x is tracked
enter ZOI
group x is inside ZOI
exit ZOI
start walking, start running
exit ZOI
stops
group x is stopped in the ZOI > 30 sec
blocking