



University of Hamburg

MIN Faculty
Department of Informatics
Cognitive Systems Laboratory



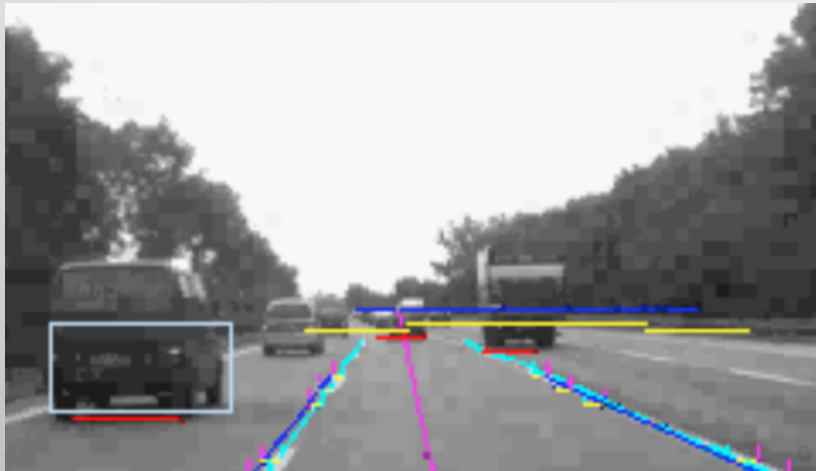
Ontology-based Realtime Activity Monitoring Using Beam Search

Wilfried Bohlken, Bernd Neumann
University of Hamburg

Lothar Hotz
HITEC

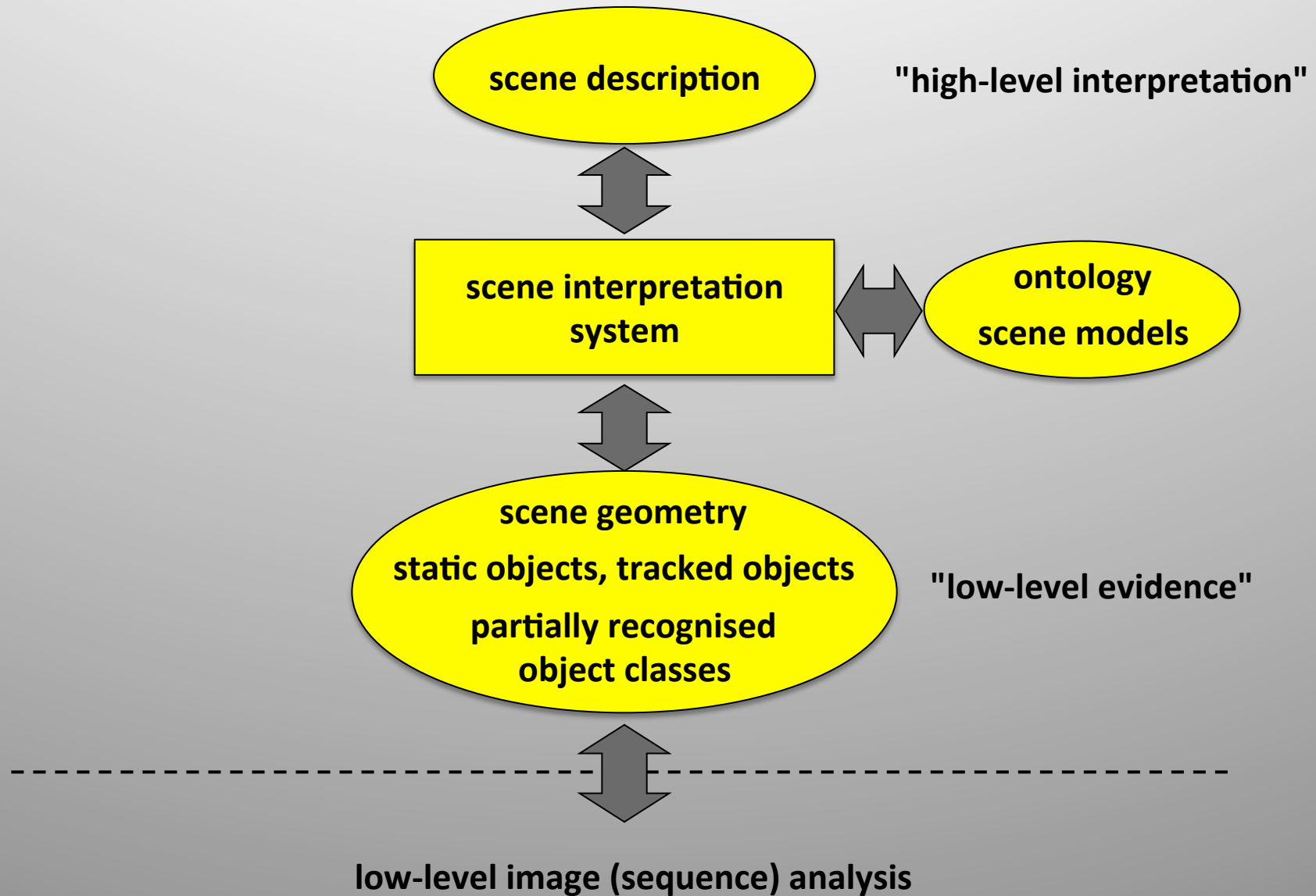
Patrick Koopmann
Cirquent GmbH

Activity Recognition has Numerous Applications



Generic architecture for scene interpretation?

Scope: Knowledge-based Scene Interpretation



Aircraft Turnaround Monitoring

- **Recognition of multi-object activities such as**
 - Aircraft Arrival Preparation
 - Passenger Ramp Motion
 - Unloading
 - Loading
 - Refuelling
 - Aircraft Departure
- **Recognition of complete turnarounds**
 - Monitoring of temporal constraints
 - Monitoring of unusual activities
- **Large variability of subtasks and turnarounds**
- **Large number of unrelated activities**
- **Uncontrolled environment, difficult low-level image analysis**



Requirements for a Generic Solution

- Incremental real-time recognition
 - ➔ Parallel processing of multiple partial interpretations
- Preference measure for resolving ambiguities
 - ➔ Context-dependent probabilistic rating
- Image analysis for uncontrolled real-world domains
 - ➔ Dealing with missing and erroneous evidence
- Knowledge-based architecture with reusable knowledge base
 - ➔ OWL-DL ontology with SWRL rules

Related Work

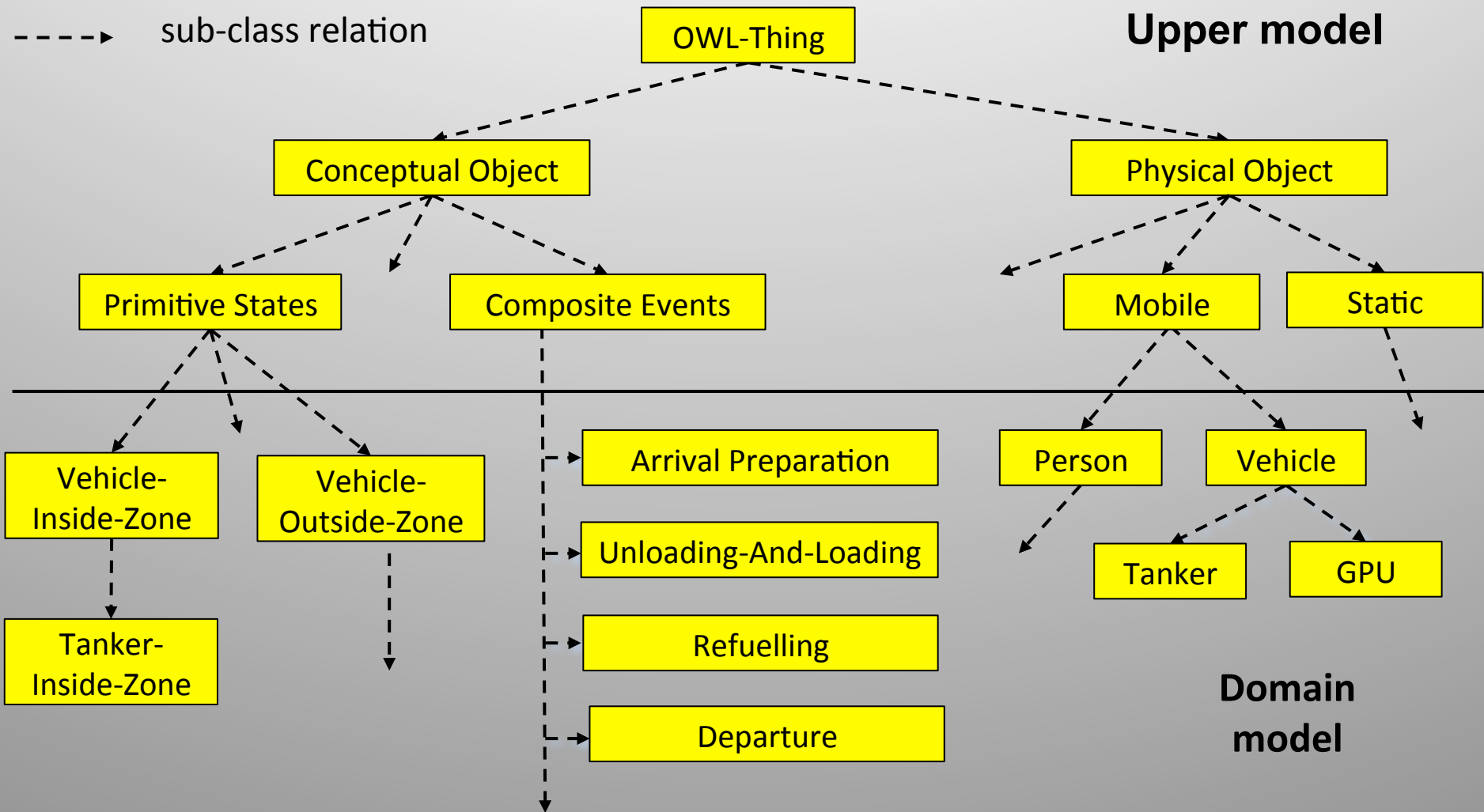
- **Badler 1975 (conceptual descriptions of object motions)**
- **Neumann 1989 (natural language description of of traffic scenes)**
- **Rimey 1993 (Bayesian networks for vision control)**
- **Nagel 1999 (situation graph trees)**
- **Thonnat, Brémond 2007 (scenario recognition)**
- **Zhu & Mumford 2007 (stochastic grammar of images)**
- **Moeller 2010 (logic-based media interpretation)**

=> recognising hierarchical compositional structures

Representing Activity Concepts in OWL-DL

- **OWL is a standardised ontology language**
 - Definition of properties, aggregate taxonomies and partonomies
 - Knowledge editor Protégé in wide use
- **Powerful Description Logic reasoners support OWL-DL**
 - Useful services for large high-level knowledge bases
 - No support for stepwise recognition
 - No support for constraint solving
- **Crisp relations**
 - Fuzzy or probabilistic information cannot be represented
- **SWRL extension for rules**
 - Awkward definition of quantitative constraints

Taxonomy for Turnaround Activities



Using the Protégé Editor

Concepts are defined with taxonomical and binary relations (roles)

The screenshot displays the Protégé Editor interface. On the left, a class hierarchy is shown with a tree view. The root class is 'Thing', which is expanded to show its subclasses: 'ap-upper:Constant', 'Scene-Object', 'Conceptual-Object', 'Event', 'State', and 'Physical-Object'. Under 'Event', there are 'Composite-Event' and 'Primitive-Event'. Under 'Primitive-Event', the class 'Vehicle-Enters-Zone' is selected and highlighted with a yellow border. On the right, the 'Roles' panel is visible, showing a list of roles for the selected class. The roles are: 'Primitive-Event', 'has-part-vehicle-inside-zone exactly 1 Vehicle-Inside-Zone', 'has-part-vehicle-outside-zone exactly 1 Vehicle-Outside-Zone', 'has-part-time-point-begin exactly 1 Time-Point-Begin', and 'has-part-time-point-end exactly 1 Time-Point-End'. The roles are listed in a table-like structure with a yellow background for the selected class.

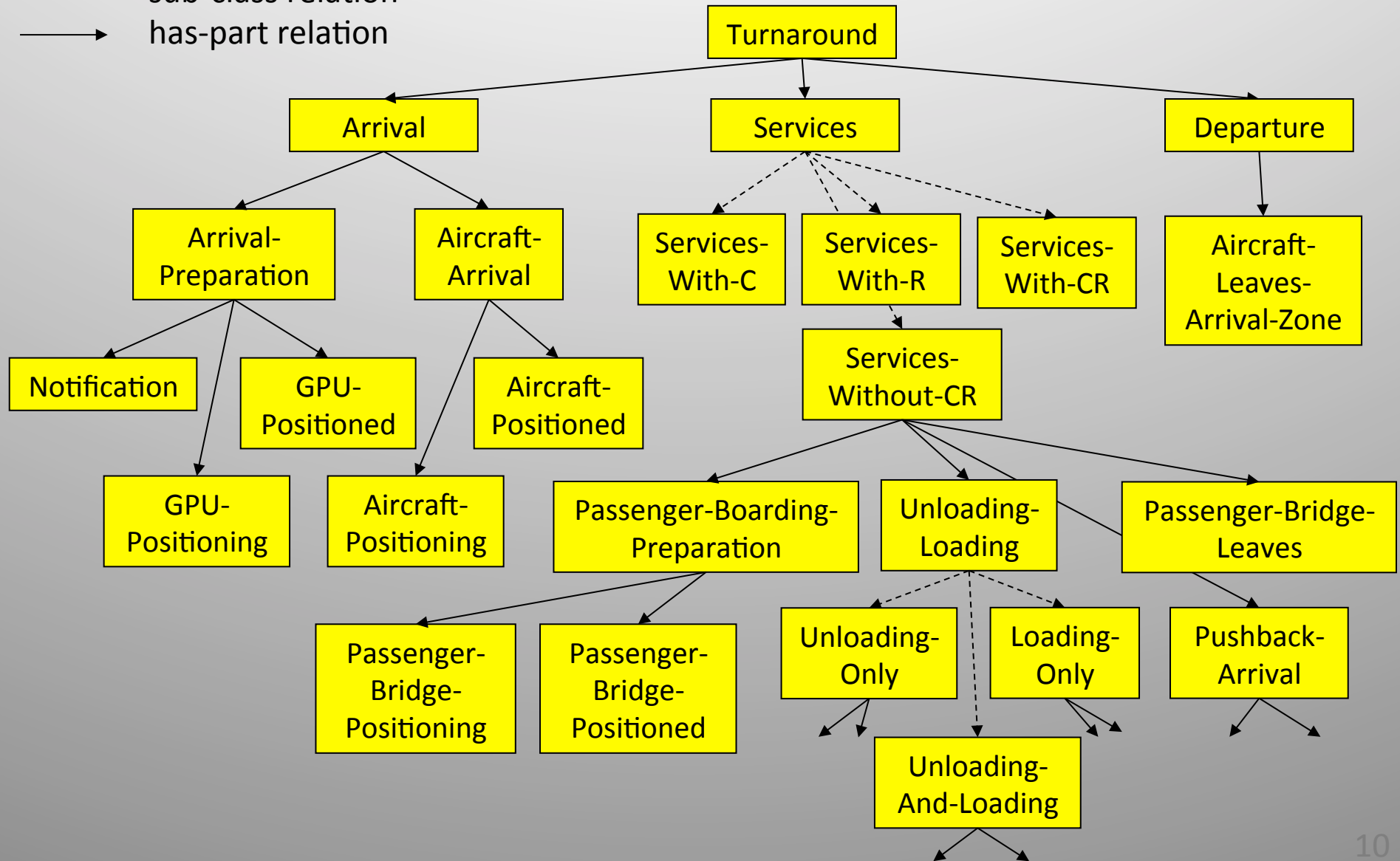
- Thing
 - ap-upper:Constant
 - ap-upper:Integer-Constant
 - Scene-Object
 - Conceptual-Object
 - Event
 - Composite-Event
 - Primitive-Event
 - Vehicle-Enters-Zone
 - State
 - Composite-State
 - Primitive-State
 - Physical-Object

Roles for Vehicle-Enters-Zone:

- Primitive-Event
- has-part-vehicle-inside-zone **exactly** 1 Vehicle-Inside-Zone
- has-part-vehicle-outside-zone **exactly** 1 Vehicle-Outside-Zone
- has-part-time-point-begin **exactly** 1 Time-Point-Begin
- has-part-time-point-end **exactly** 1 Time-Point-End

Compositional Hierarchy

-----> sub-class relation
 ———> has-part relation



Temporal Constraints in OWL

Monitoring service activities requires quantitative temporal constraints.

Passenger stairs must be positioned not later than 5 minutes after aircraft arrival.

A GPU will stop not later than 1 minute after entering the GPU zone.

In OWL, quantitative constraints can only be represented using the rule extension SWRL or – in OWL 2 – using OWL-RL.

SWRL rules have disadvantages:

- **Not elegantly connected to OWL classes**
- **Reasoning with SWRL is undecidable (in general)**

Example of Temporal SWRL Rule

OWL class definition of a vehicle visiting a zone

$\text{Visit} \sqsubseteq \text{Composite-Event} \sqcap$
 $\text{has-part1 exactly 1 Vehicle-Enters-Zone} \sqcap$
 $\text{has-part2 exactly 1 Vehicle-Leaves-Zone}$

**"Visit begins with Vehicle-Enters-zone and ends with Vehicle-Leaves-Zone.
Vehicle-Enters-Zone and Vehicle-Leaves-Zone have the same agent and zone, respectively."**

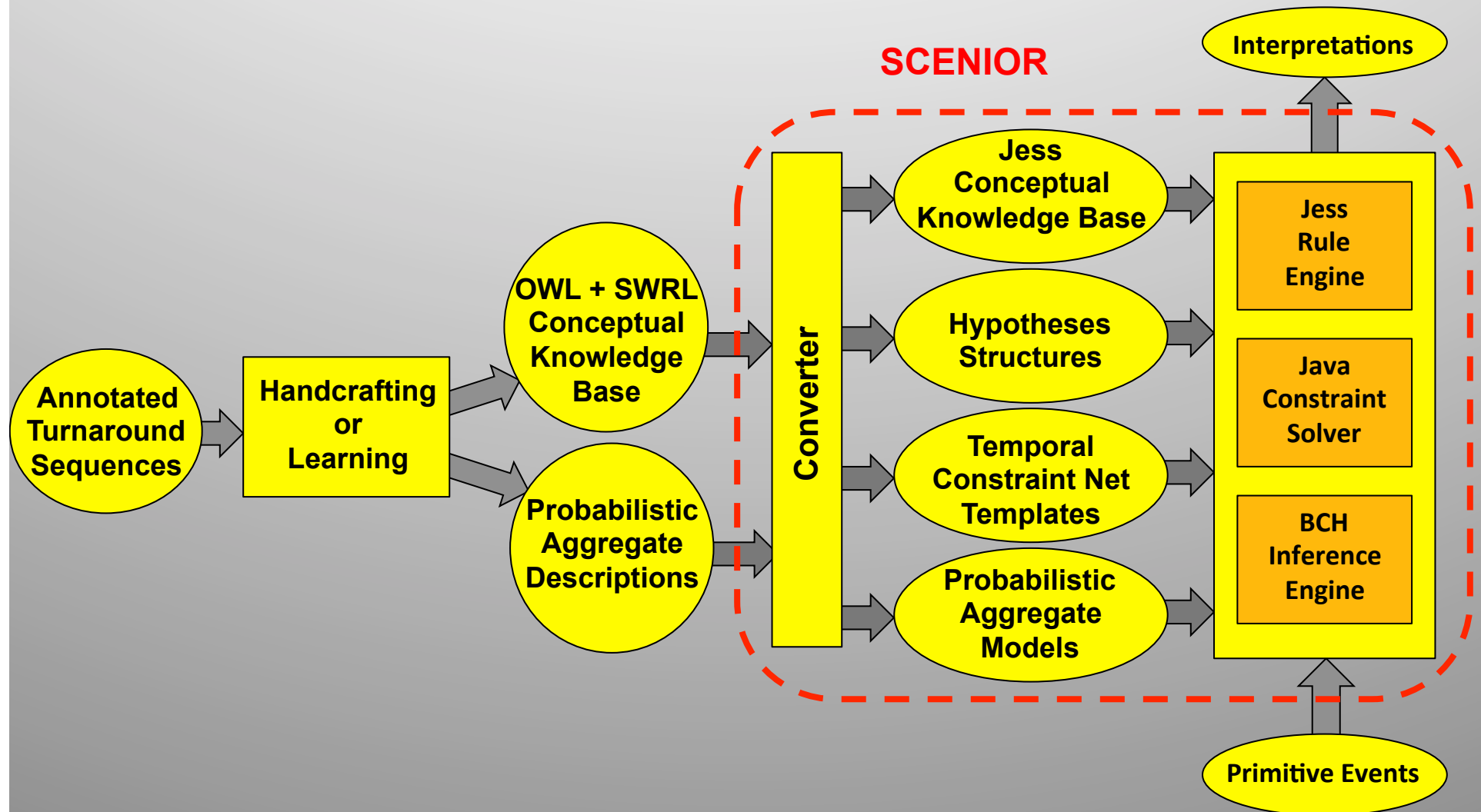
SWRL rule premise establishes variable names

$\text{Visit}(\text{?vis})$
 $\wedge \text{has-part1}(\text{?vis}, \text{?veh-enters})$
 $\wedge \text{has-part2}(\text{?vis}, \text{?veh-leaves})$
 $\wedge \text{has-start-time}(\text{?vis}, \text{?vis-st})$
 $\wedge \text{has-finish-time}(\text{?vis}, \text{?vis-ft})$
 $\wedge \text{has-time-point}(\text{?veh-enters}, \text{?veh-enters-tp})$
 $\wedge \text{has-agent}(\text{?veh-enters}, \text{?veh-enters-ag})$
 $\wedge \text{has-zone}(\text{?veh-enters}, \text{?veh-enters-zn})$
 $\wedge \text{has-time-point}(\text{?veh-leaves}, \text{?veh-leaves-tp})$
 $\wedge \text{has-agent}(\text{?veh-leaves}, \text{?veh-leaves-ag})$
 $\wedge \text{has-zone}(\text{?veh-leaves}, \text{?veh-leaves-zn})$

SWRL rule consequence specifies identity constraints and temporal constraints

\rightarrow
 $\text{?vis-st} = \text{?veh-enters-tp}$
 $\wedge \text{?vis-ft} = \text{?veh-leaves-ft}$
 $\wedge \text{?veh-enters-ag} = \text{?veh-leaves-ag}$
 $\wedge \text{?veh-enters-zn} = \text{?veh-leaves-zn}$
 $\wedge \text{?veh-enters-tp} \leq \text{?veh-leaves-tp}$

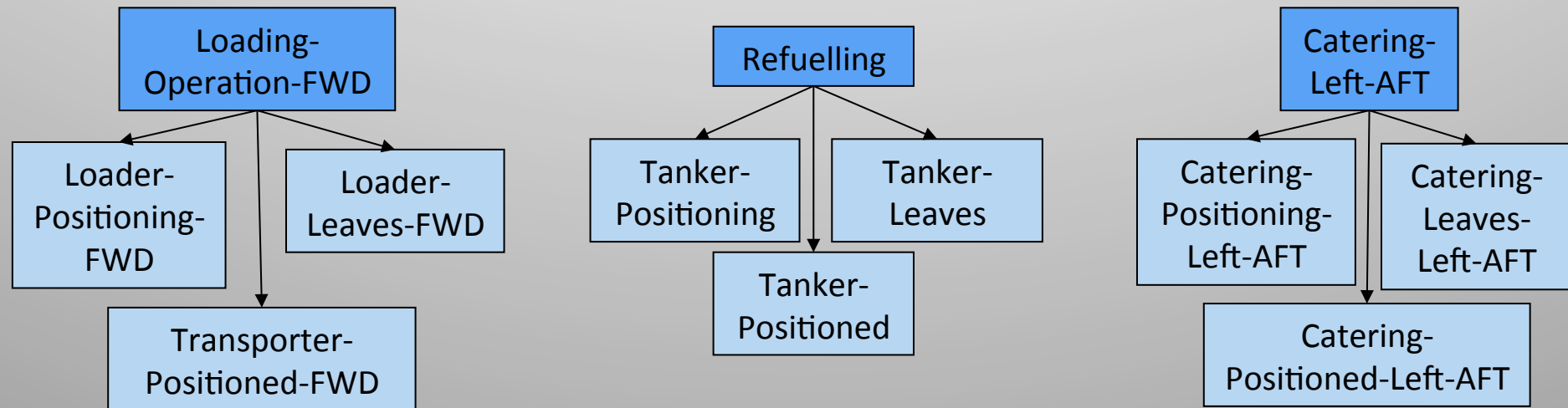
Transforming an OWL Knowledge Base into an Operational Interpretation System



Generating Hypotheses Structures for the JESS Working Memory (1)

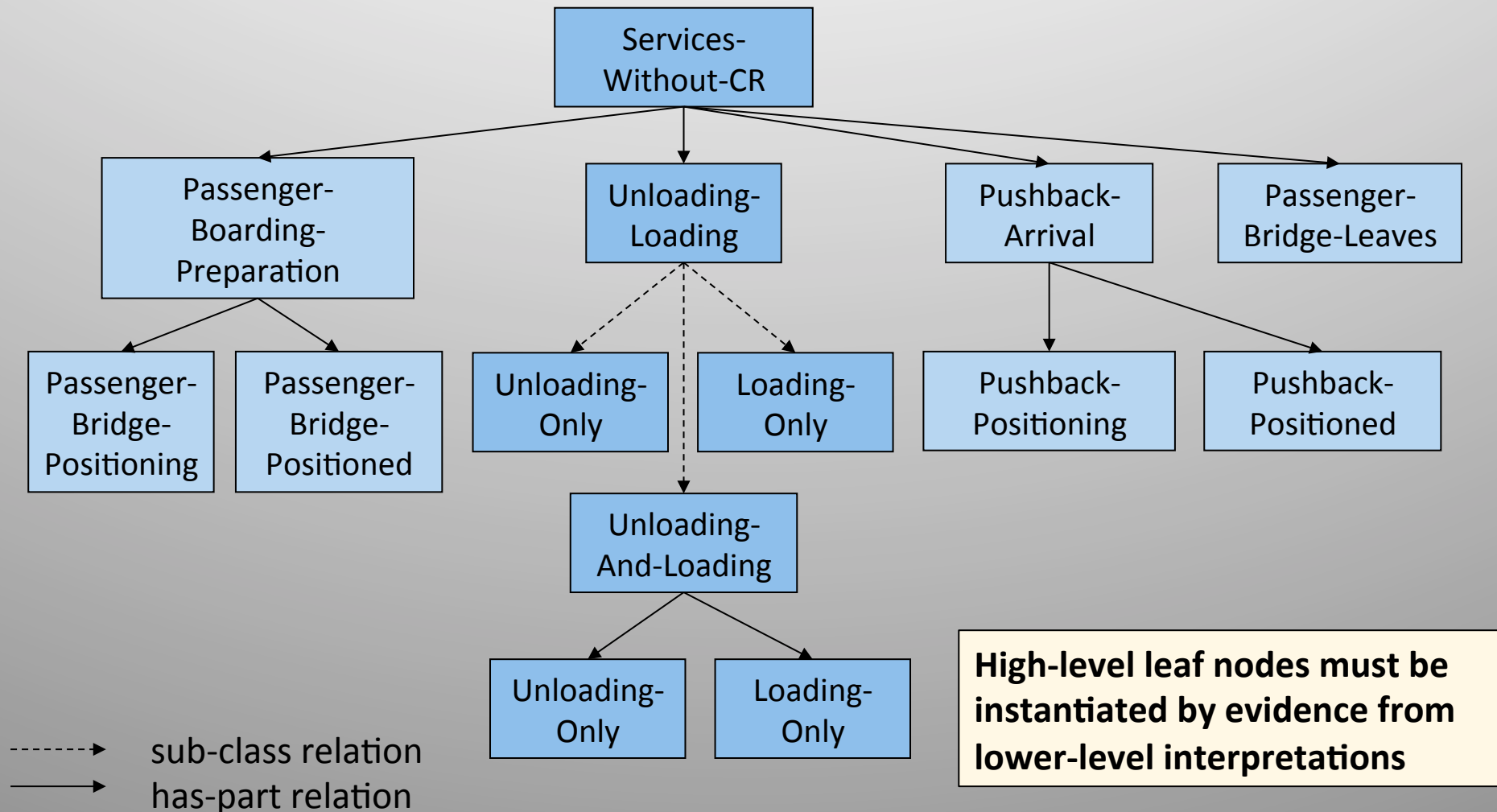
Hypotheses structures provide independent interpretation goals:

- Basis for prediction and ranking
- Single representation for multiple or alternative occurrences
- Certain parts may be marked as hallucinatable

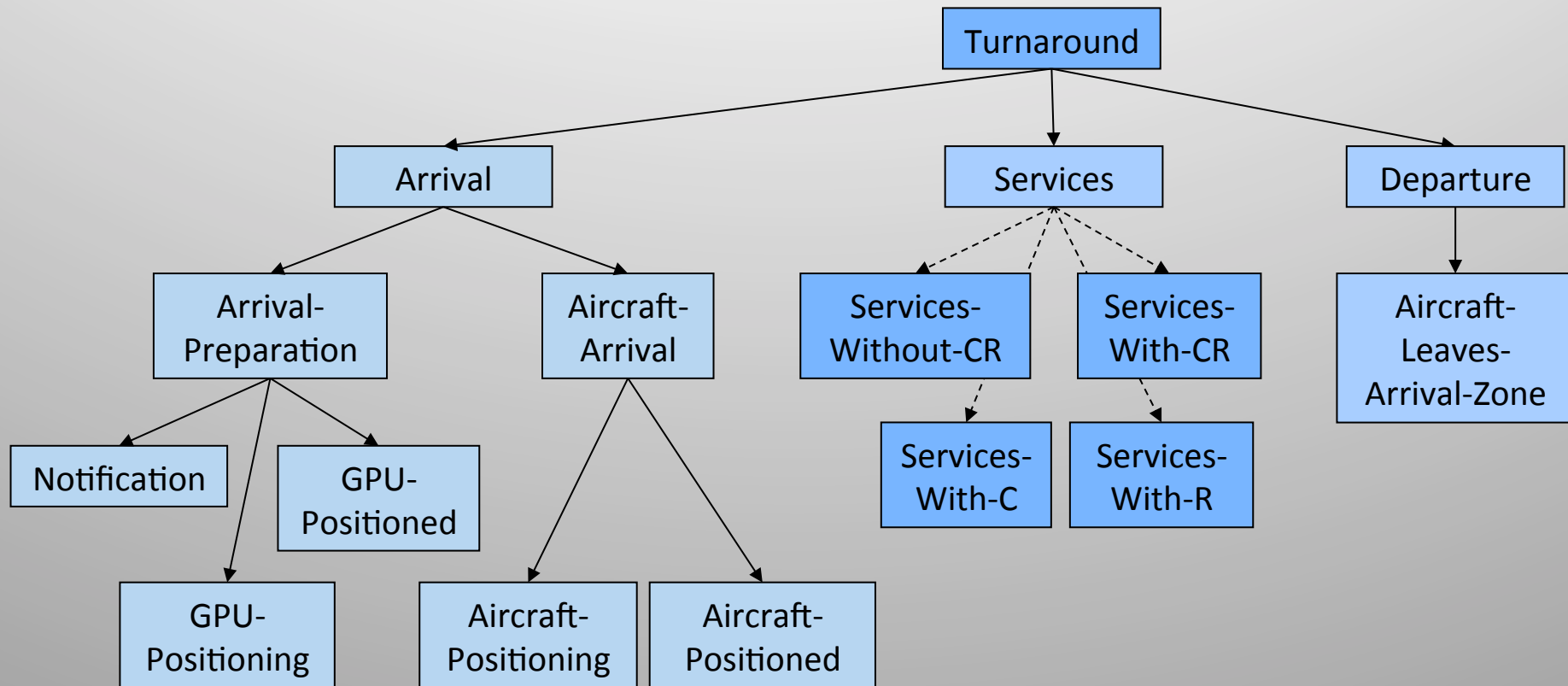


Low-level leaf nodes must be instantiated by evidence from low-level image analysis

Generating Hypotheses Structures for the JESS Working Memory (2)



Generating Hypotheses Structures for the JESS Working Memory (3)

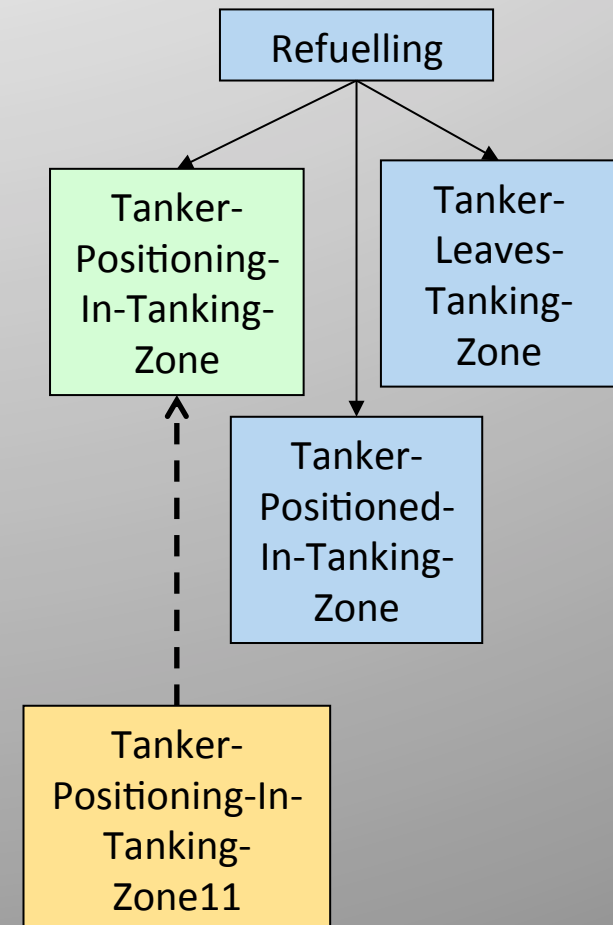


-----> sub-class relation
-----> has-part relation

Generating JESS Interpretation Rules (1)

Evidence-assignment rule for compositional leaf nodes

```
(defrule Refuelling_ea_rule
  ?e-id <- (Tanker-Positioning-In-Tanking-Zone
            (name ?tpg_17)
            (status evidence))
  ?h-id <- (Tanker-Positioning-In-Tanking-Zone
            (name ?tpg_h)
            (status ?status_1))
  (test (or (eq ?status_1 hypothesised)
            (eq ?status_1 hallucinated)))
  ;; check temporal constraints
  =>
  (modify ?e-id (status assigned))
  (modify ?h-id (status instantiated))
  ;; update temporal constraint net
)
```



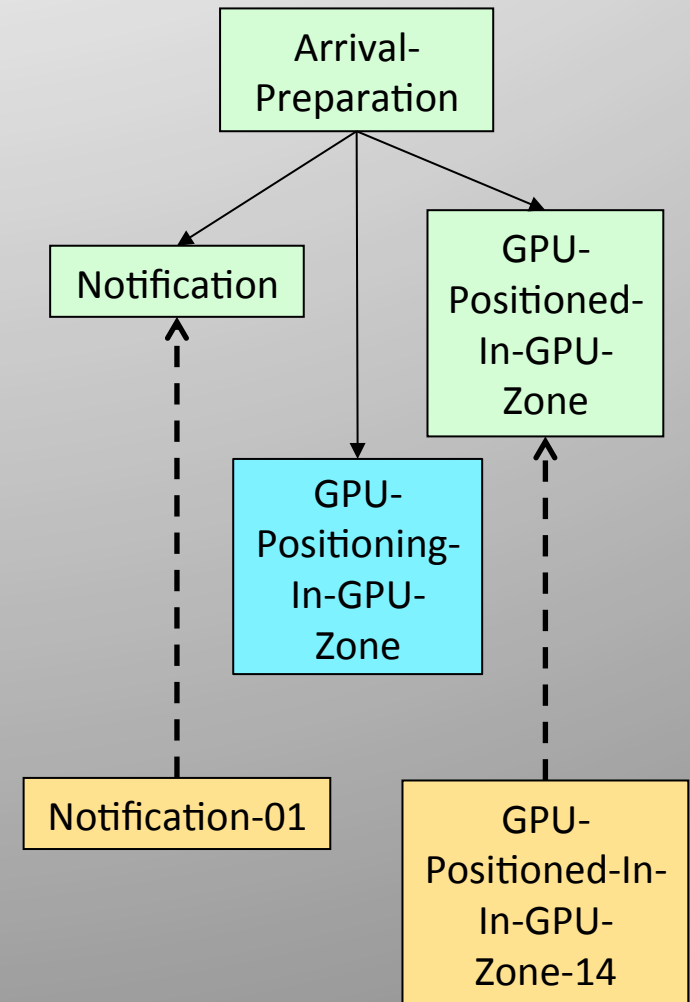
Generating JESS Interpretation Rules (2)

Aggregate-instantiation rule for aggregates

```
(defrule Arrival-preparation_ai_rule

  ?h-id <- (Arrival-Preparation (name ?ap_h)
          (status hypothesised)
          (has-part-1 p1)
          (has-part-2 p2)
          (has-part-3 p3))
  (Notification (name ?p1)
          (status ?status_1))
  (test (or (eq ?status_1 instantiated)
            (eq ?status_1 hallucinated)))
  (GPU-Positioning-In-GPU-Zone (name ?p2)
          (status ?status_2))
  (test (or (eq ?status_2 instantiated)
            (eq ?status_2 hallucinated)))
  (GPU-Positioned-In-GPU-Zone (name ?p3)
          (status ?status_3))
  (test (or (eq ?status_3 instantiated)
            (eq ?status_3 hallucinated)))

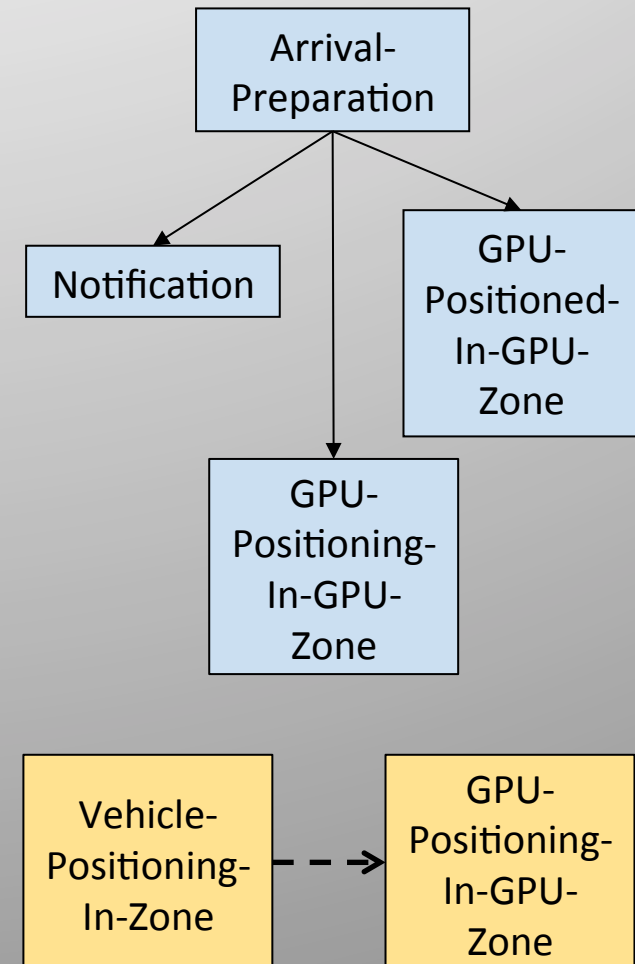
=>
  (modify ?h-id (status instantiated)))
```



Generating JESS Interpretation Rules (3)

Specialisation rule for agent or location of primitive events

```
(defrule GPU-Positioning-In-GPU-Zone_s_rule
  ?e-id <- (Vehicle-Positioning-In-Zone
            (name ?vez_14)
            (status evidence)
            (has-agent ?a1)
            (has-location ?l1))
  (GPU (name ?a1))
  (GPU-Zone (name ?l1))
  (not (GPU-Positioning-In-GPU-Zone
        (name ?vez_14)))
  =>
  (retract ?e-id)
  (assert (GPU-Positioning-In-GPU-Zone
          (name ?vez_14)
          (status evidence)
          (has-agent ?a1)
          (has-location ?l1))))
```



Beam Search with JESS

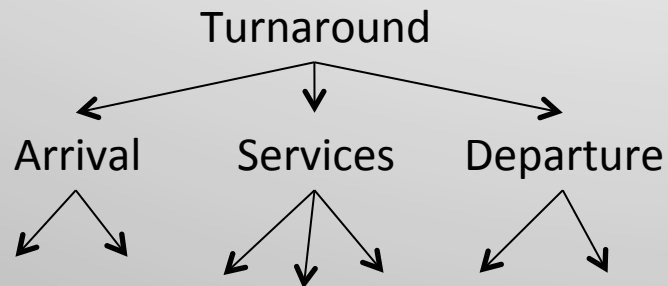
- Hypotheses structures are initialised as independent interpretation threads.
- New evidence is assigned to all matching threads or to clutter.
- Interpretation threads are cloned in case of multiple assignment possibilities.
- Low-rating threads exceeding the beam width are discarded.

Implementation in SCENIOR:

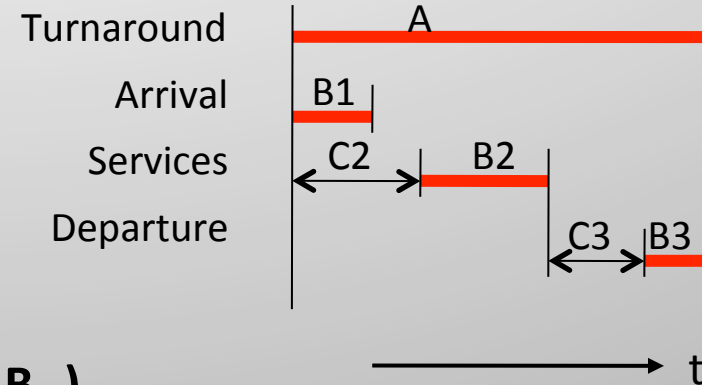
- SCENIOR can accommodate up to 100 threads.
- Ca. 800 threads are created for a typical turnaround scene.

Probabilistic Preference Measure Based on Aggregate JPDs

Aggregate partonomy



Temporal aggregate structure



Aggregate JPD $P_{\text{Turnaround}}(A B_1 C_2 B_2 C_3 B_3)$
 $\Rightarrow P'_{\text{Turnaround}}(B_1 C_2 B_2 C_3 B_3 | A)$

For **Bayesian Compositional Hierarchies** (BCHs):

Scene JPD $P_{\text{Scene}}^m = p_m P'_{\text{Turnaround}} P'_{\text{Arrival}} \dots P'_{\text{Refuelling}} \dots P'_{\text{Pushback}} P_{\text{clutter}}$

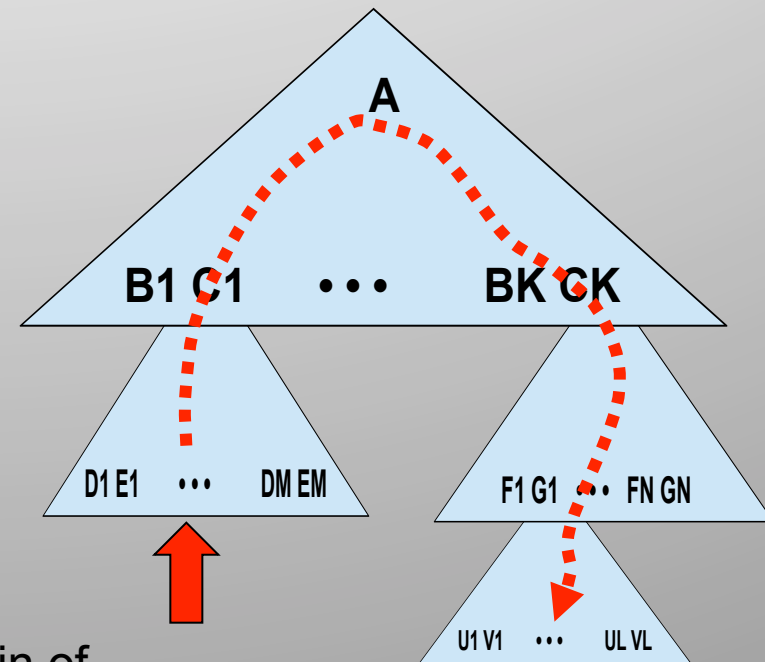
Ranking of partial interpretations with evidence $e_1 \dots e_k$: $P_{\text{Scene}}^m(e_1 \dots e_k)$

Probability Propagation

Representation of durations and offsets by Gaussians allows efficient probability update.

Bayesian Compositional Hierarchy (BCH)

- Enter begin or end of events
 - Propagate change throughout BCH
 - Estimate non-instantiated temporal variables
- ➔ obtain dynamic priors (context-dependent)

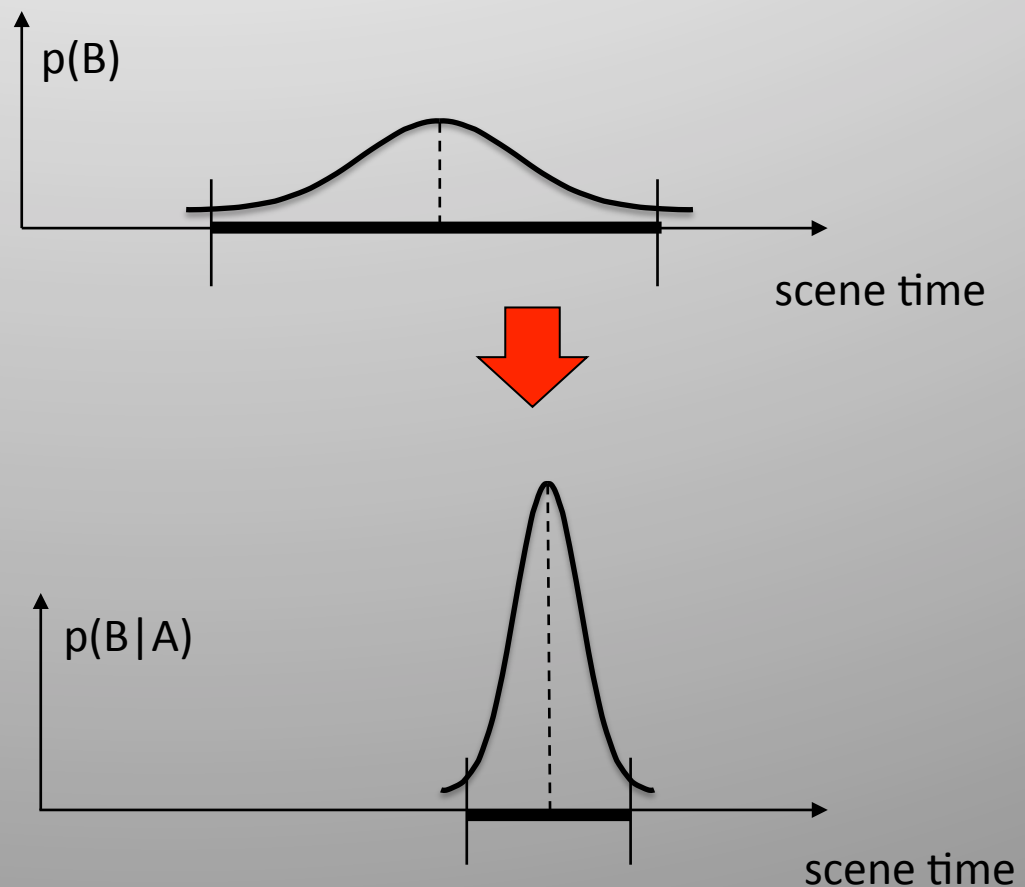


begin of
Unloading-Operation
at $t = 26$

Dynamic Priors for Multivariate Gaussians

Gaussian multivariate distributions allow highly efficient probability propagation.

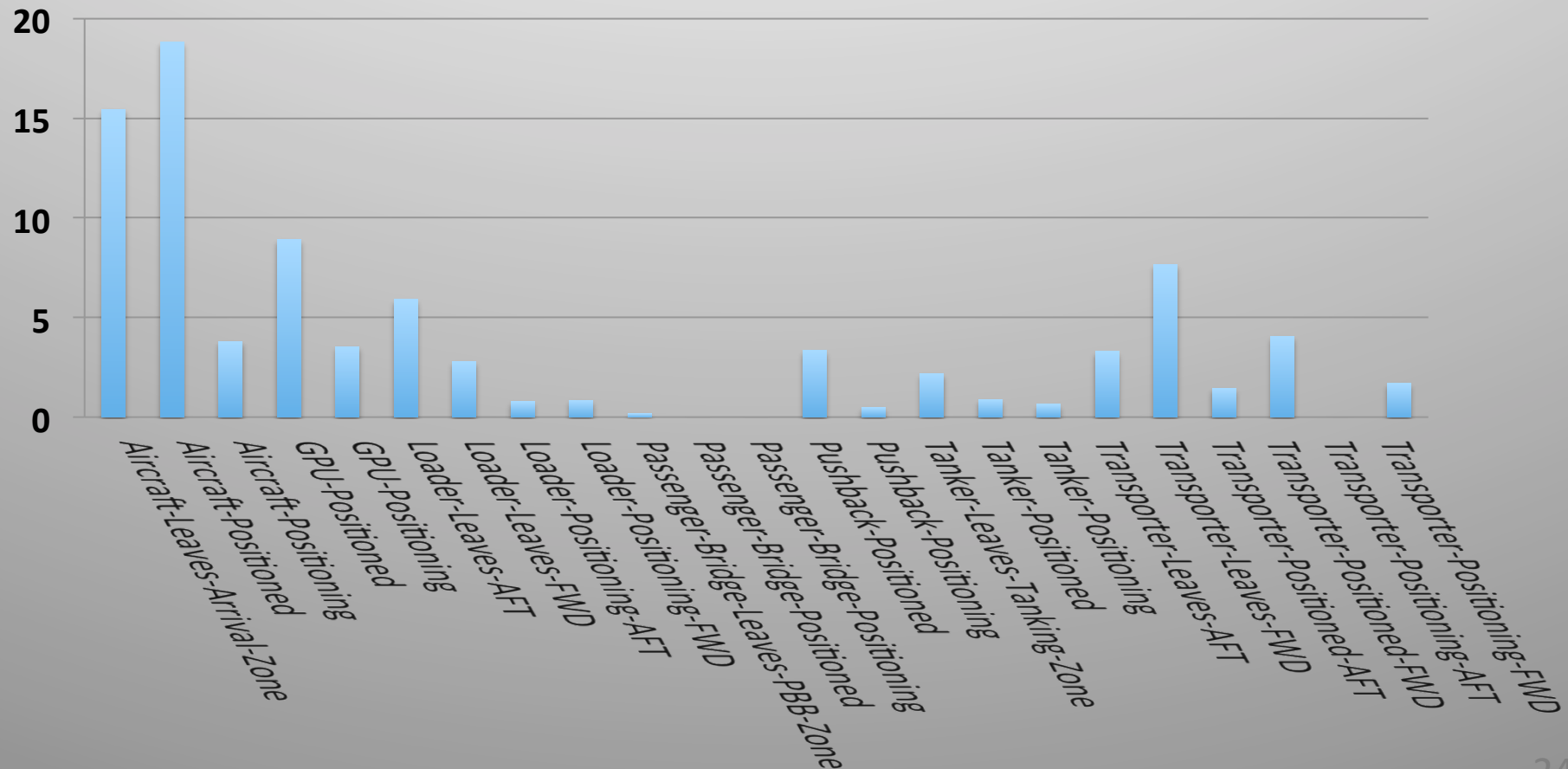
Expected temporal distribution of event B is changed by propagating observed time of event A



Experiments: False Positives

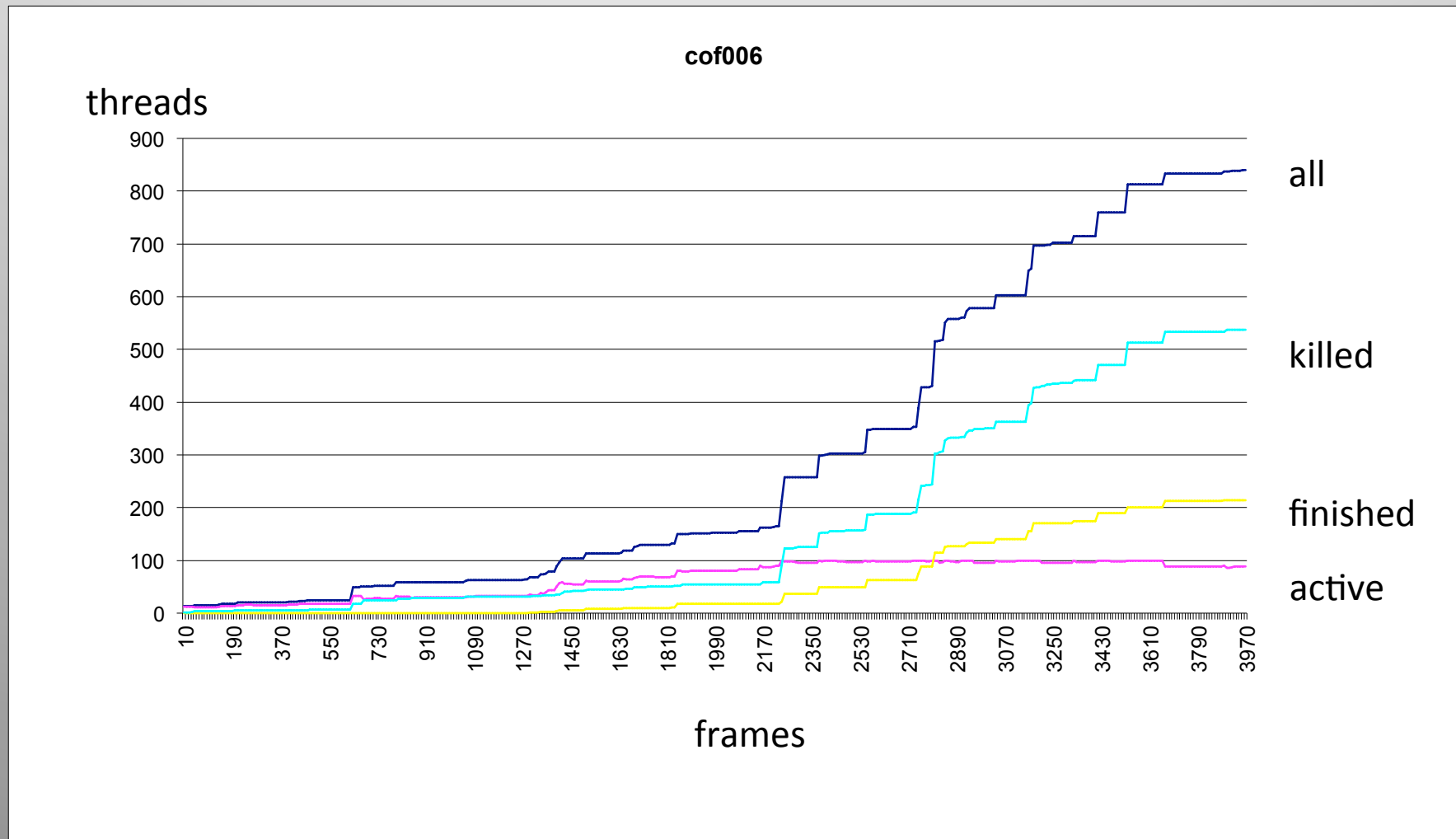
Low-level image analysis noise and unrelated scene activities have caused a large number of false positives.

➔ Positive evidence is always interpreted both as turnaround and clutter.



Thread Statistics

Low input data quality requires the use of full beam width (100 threads).



Recognition Results

- Models trained on 32 annotated turnaround sequences, tested on 20 other sequences
- Complete turnarounds recognised for all but 3 highly irregular sequences
- 75% of all activities "correctly" recognised (overlap with annotated interval)

SEQUENCE	1	2	3	4	5	6	8	9	16	18	25	29	58	59	62	65	66
Arrival	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
Passenger-Boarding-Preparation	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Unloading-Loading-AFT	1	1	1	0	1	0	0	1	0	1	0	0	0	0	1	0	0
Unloading-Loading-FWD				1		1	1	0					1	1		1	1
Refuelling			0	1	0			0			0		0		1	1	
Pushback-Arrival	1	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0
Passenger-Bridge-Leaves-PBB-Zone	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Departure	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Turnaround Interpretation Log: Notification

SCENIOR [Window Title Bar]

File Options Run About

Ontology: C:\Co-Friend\Source\Scenior\Ontologies\model-V14.owl

Data file: C:\Co-Friend\Source\Scenior\Data\Set_2011-03-01-SVT-fusion\cof062.txt

Dataset: [Dropdown]

Processing Primitives Log

Processing [Buttons: Play, Refresh, Stop, Pause] Processing [Indicator] Frame: 5 Entries in queue: 2283 model [Dropdown] Threads: 1 Active: 1 Finished: 0 Dead: 0

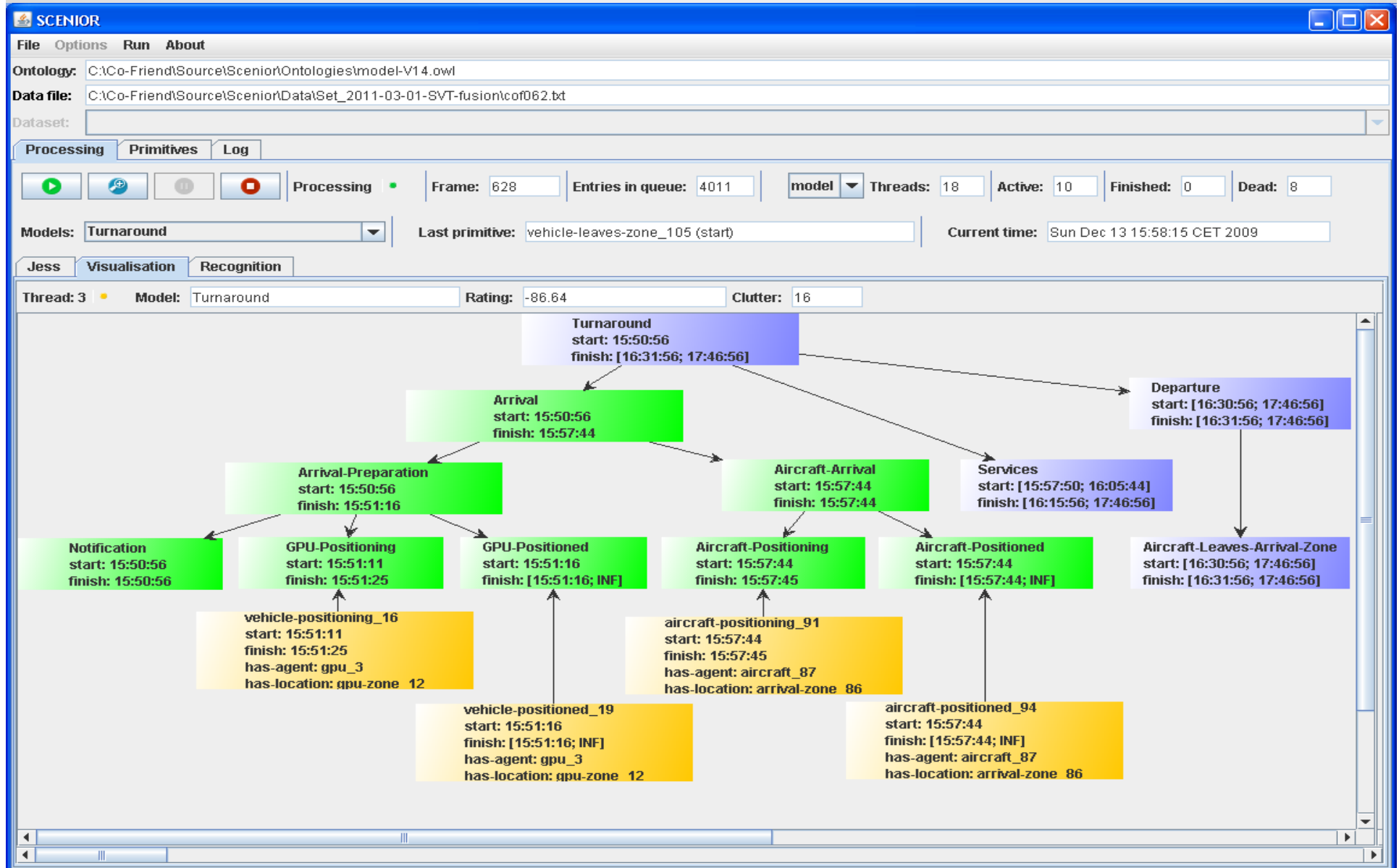
Models: Turnaround [Dropdown] Last primitive: vehicle-leaves-zone_7 (start) Current time: Sun Dec 13 15:50:59 CET 2009

Jess Visualisation Recognition

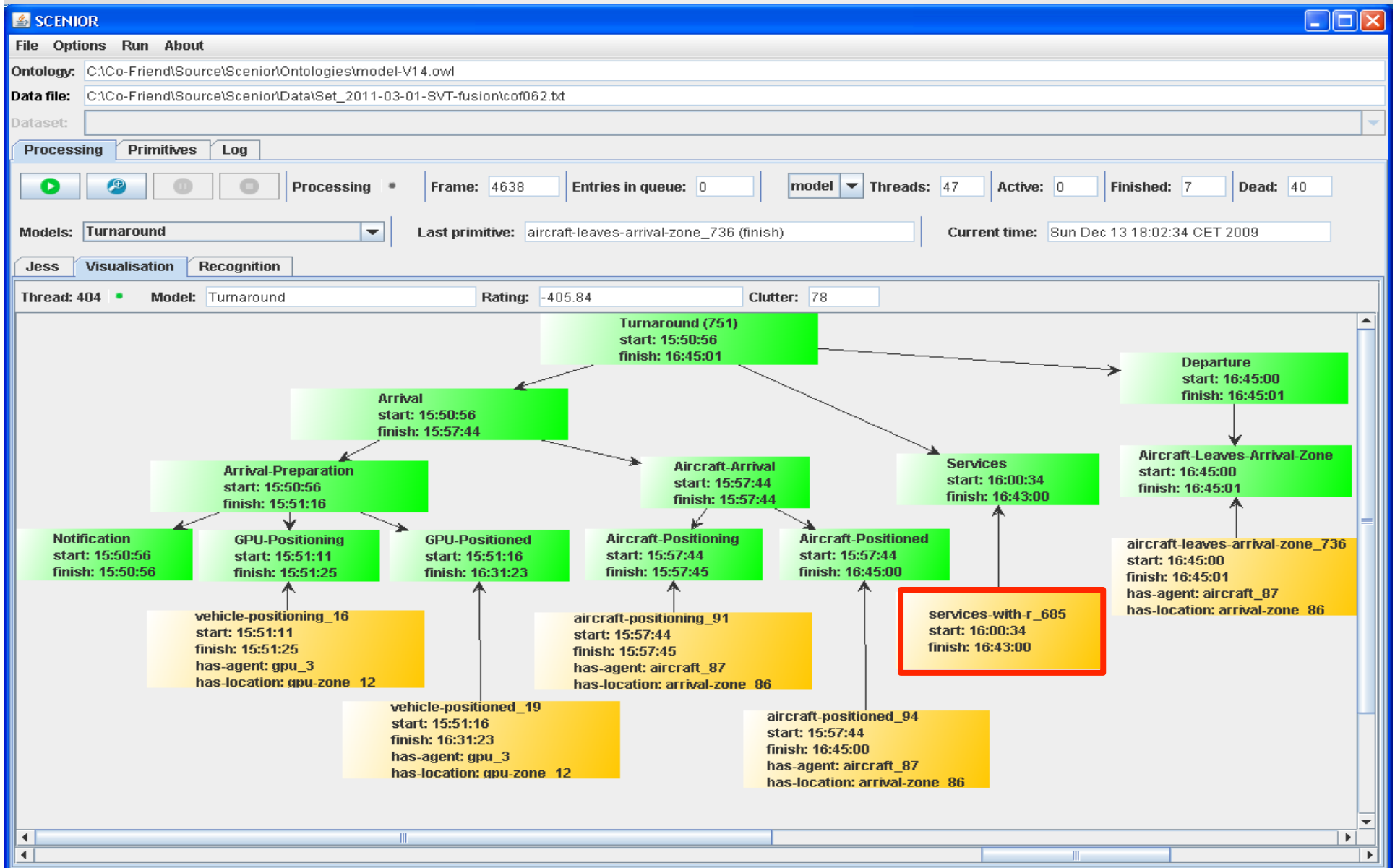
Thread: 3 Model: Turnaround Rating: -9.21 Clutter: 1

```
graph TD; Turnaround["Turnaround  
start: 15:50:56  
finish: [16:31:56; 17:46:56]"] --> Arrival["Arrival  
start: 15:50:56  
finish: [15:51:56; 16:50:56]"]; Turnaround --> Departure["Departure  
start: [16:30:56; 17:46:56]  
finish: [16:31:56; 17:46:56]"]; Arrival --> ArrivalPrep["Arrival-Preparation  
start: 15:50:56  
finish: [15:50:56; 16:14:56]"]; Arrival --> AircraftArrival["Aircraft-Arrival  
start: [15:50:56; 16:50:56]  
finish: [15:51:56; 16:50:56]"]; ArrivalPrep --> Notification["Notification  
start: 15:50:56  
finish: 15:50:56"]; ArrivalPrep --> GPUPositioning["GPU-Positioning  
start: [15:50:56; 16:13:56]  
finish: [15:50:56; 16:14:56]"]; ArrivalPrep --> GPUPositioned["GPU-Positioned  
start: [15:50:56; 16:14:56]  
finish: [15:50:56; INF]"]; AircraftArrival --> AircraftPositioning["Aircraft-Positioning  
start: [15:50:56; 16:50:56]  
finish: [15:50:56; 16:51:56]"]; AircraftArrival --> AircraftPositioned["Aircraft-Positioned  
start: [15:51:56; 16:50:56]  
finish: [15:51:56; INF]"]; Departure --> Services["Services  
start: [15:52:02; 16:58:56]  
finish: [16:15:56; 17:46:56]"]; Departure --> AircraftLeavesZone["Aircraft-Leaves-Arrival-Zone  
start: [16:30:56; 17:46:56]  
finish: [16:31:56; 17:46:56]"];
```

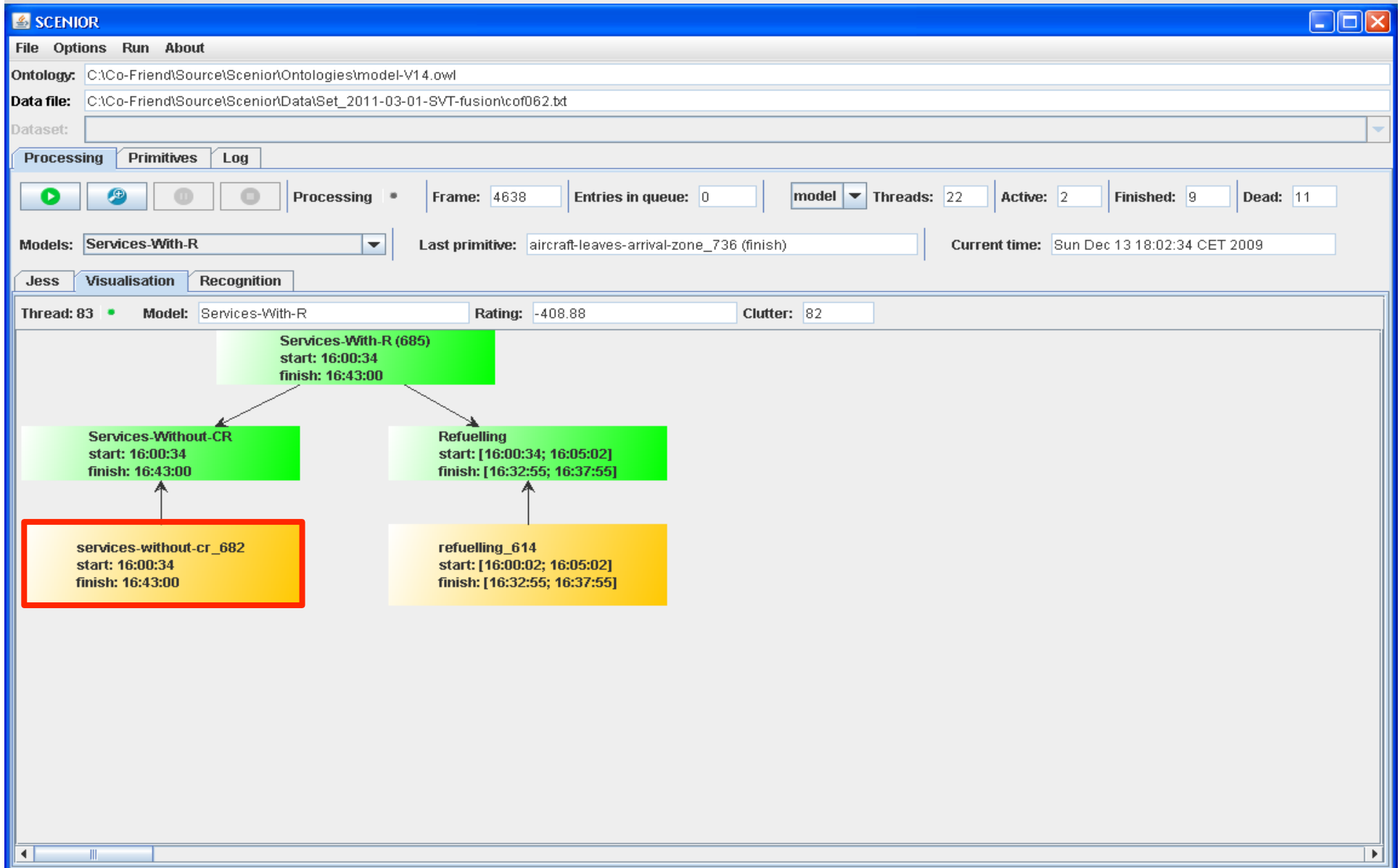
Turnaround Interpretation Log: Arrival



Turnaround Interpretation Log: Complete



Turnaround Interpretation Log: Services with Refuelling



Turnaround Interpretation Log: Services without Catering and Refuelling

SCENIOR [Window Title]

File Options Run About

Ontology: C:\Co-Friend\Source\Scenior\Ontologies\model-V14.owl

Data file: C:\Co-Friend\Source\Scenior\Data\Set_2011-03-01-SVT-fusion\cof062.bt

Dataset: [Dropdown]

Processing Primitives Log

Processing [Buttons] Frame: 4638 Entries in queue: 0 model Threads: 188 Active: 1 Finished: 9 Dead: 178

Models: Services-Without-CR Last primitive: aircraft-leaves-arrival-zone_736 (finish) Current time: Sun Dec 13 18:02:34 CET 2009

Jess Visualisation Recognition

Thread: 314 Model: Services-Without-CR Rating: -404.86 Clutter: 83

```

graph TD
    Root["Services-Without-CR (682)  
start: 16:00:34  
finish: 16:43:00"]
    Root --> PBP["Passenger-Boarding-Preparation  
start: 16:00:34  
finish: 16:00:34"]
    Root --> PA["Pushback-Arrival  
start: [16:00:34; 16:43:00]  
finish: [16:00:34; 16:43:00]"]
    Root --> UL["Unloading-Loading  
start: 16:03:07  
finish: [16:33:56; 16:36:56]"]
    Root --> PBLPZ["Passenger-Bridge-Leaves-PBB-Zone  
start: 16:42:59  
finish: 16:43:00"]
    
    PBP --> PBP_Pos["Passenger-Bridge-Positioning  
start: 16:00:34  
finish: 16:00:35"]
    PBP --> PBP_Pos_2["Passenger-Bridge-Positioned  
start: 16:00:34  
finish: 16:42:59"]
    
    UL --> UL_572["unloading-and-loading_572  
start: 16:03:07  
finish: [16:33:56; 16:36:56]"]
    UL --> PBLPZ_679["passenger-bridge-leaves-pbb-zone_679  
start: 16:42:59  
finish: 16:43:00  
has-agent: passenger-boarding-bridge_678  
has-location: pbb-zone_137"]
    
    PBP_Pos --> PBP_Pos_138["passenger-bridge-positioning_138  
start: 16:00:34  
finish: 16:00:35  
has-agent: passenger-boarding-bridge_136  
has-location: pbb-zone_137"]
    PBP_Pos_2 --> PBP_Pos_141["passenger-bridge-positioned_141  
start: 16:00:34  
finish: 16:42:59  
has-agent: passenger-boarding-bridge_136  
has-location: pbb-zone_137"]
    
    PBLPZ_679 --> PBLPZ_678["passenger-bridge-leaves-pbb-zone_678  
start: 16:42:59  
finish: 16:43:00  
has-agent: passenger-boarding-bridge_678  
has-location: pbb-zone_137"]
  
```

Thread: 314 Model: Services-Without-CR Rating: -404.86 Clutter: 83

Services-Without-CR (682)
start: 16:00:34
finish: 16:43:00

Passenger-Boarding-Preparation
start: 16:00:34
finish: 16:00:34

Pushback-Arrival
start: [16:00:34; 16:43:00]
finish: [16:00:34; 16:43:00]

Unloading-Loading
start: 16:03:07
finish: [16:33:56; 16:36:56]

Passenger-Bridge-Leaves-PBB-Zone
start: 16:42:59
finish: 16:43:00

Passenger-Bridge-Positioning
start: 16:00:34
finish: 16:00:35

Passenger-Bridge-Positioned
start: 16:00:34
finish: 16:42:59

unloading-and-loading_572
start: 16:03:07
finish: [16:33:56; 16:36:56]

passenger-bridge-leaves-pbb-zone_679
start: 16:42:59
finish: 16:43:00
has-agent: passenger-boarding-bridge_678
has-location: pbb-zone_137

passenger-bridge-positioning_138
start: 16:00:34
finish: 16:00:35
has-agent: passenger-boarding-bridge_136
has-location: pbb-zone_137

passenger-bridge-positioned_141
start: 16:00:34
finish: 16:42:59
has-agent: passenger-boarding-bridge_136
has-location: pbb-zone_137

Turnaround Interpretation Log: Unloading and Loading

The screenshot displays the SCENIOR software interface. At the top, the title bar reads "SCENIOR". Below it, the menu bar includes "File", "Options", "Run", and "About". The main window contains several fields and tabs:

- Ontology:** C:\Co-Friend\Source\Scenior\Ontologies\model-V14.owl
- Data file:** C:\Co-Friend\Source\Scenior\Data\Set_2011-03-01-SVT-fusion\cof062.txt
- Dataset:** (empty dropdown)
- Processing** (selected tab)
- Primitives** (tab)
- Log** (tab)

Below the tabs, there are control buttons (play, refresh, stop, pause) and a "Processing" indicator. The status bar shows:

- Frame: 4638
- Entries in queue: 0
- model (dropdown)
- Threads: 95
- Active: 13
- Finished: 64
- Dead: 18

The "Models" section shows "Unloading-And-Loading" selected. The "Last primitive" is "aircraft-leaves-arrival-zone_736 (finish)". The "Current time" is "Sun Dec 13 18:02:34 CET 2009".

At the bottom, the "Jess" tab is active, showing a "Visualisation" of the log. The visualization displays a tree structure of operations:

- Thread: 132**, **Model: Unloading-And-Loading**, **Rating: -378.72**, **Clutter: 80**
- Unloading-And-Loading (572)**
 - start: 16:03:07
 - finish: [16:33:56; 16:36:56]
 - Unloading-Only
 - start: 16:03:07
 - finish: [16:03:53; 16:06:01]
 - unloading-aft-and-fwd_309
 - start: 16:03:07
 - finish: [16:03:53; 16:06:01]
 - Loading-Only
 - start: [16:26:33; 16:31:33]
 - finish: [16:33:56; 16:36:56]
 - loading-fwd_569
 - start: [16:26:33; 16:31:33]
 - finish: [16:33:56; 16:36:56]

Turnaround Interpretation Log: Loading Forward

The screenshot displays the SCENIOR software interface. At the top, the title bar reads 'SCENIOR'. Below it is a menu bar with 'File', 'Options', 'Run', and 'About'. The main area contains several fields: 'Ontology: C:\Co-Friend\Source\Scenior\Ontologies\model-V14.owl', 'Data file: C:\Co-Friend\Source\Scenior\Data\Set_2011-03-01-SVT-fusion\cof062.bt', and an empty 'Dataset:' field. Below these are tabs for 'Processing', 'Primitives', and 'Log'. The 'Processing' tab is active, showing a 'Processing' button, a 'Frame: 4638' field, an 'Entries in queue: 0' field, a 'model' dropdown, 'Threads: 6', 'Active: 1', 'Finished: 4', and 'Dead: 1' fields. Below this is a 'Models:' dropdown set to 'Loading-FWD', a 'Last primitive:' field containing 'aircraft-leaves-arrival-zone_736 (finish)', and a 'Current time:' field showing 'Sun Dec 13 18:02:34 CET 2009'. At the bottom, there are tabs for 'Jess', 'Visualisation', and 'Recognition'. The 'Visualisation' tab is active, showing a thread summary: 'Thread: 279', 'Model: Loading-FWD', 'Rating: -390.02', and 'Clutter: 84'. The main visualization area shows a hierarchical diagram of tasks. At the top is a green box labeled 'Loading-FWD (569)' with 'start: [16:26:33; 16:31:33]' and 'finish: [16:33:56; 16:36:56]'. Two arrows point down from this box to two other boxes: a green box on the left labeled 'Loading-Operation-FWD' with 'start: [16:26:33; 16:31:33]' and 'finish: 16:33:56', and a cyan box on the right labeled 'Loader-Leaves-FWD' with 'start: [16:33:56; 16:35:56]' and 'finish: [16:33:56; 16:36:56]'. An arrow points up from a yellow box at the bottom left labeled 'loading-operation-fwd_555' with 'start: [16:26:33; 16:31:33]' and 'finish: 16:33:56' to the 'Loading-Operation-FWD' box. The yellow box is highlighted with a red border.

Turnaround Interpretation Log: Loading Operation Forward

SCENIOR

File Options Run About

Ontology: C:\Co-Friend\Source\Scenior\Ontologies\model-V14.owl

Data file: C:\Co-Friend\Source\Scenior\Data\Set_2011-03-01-SVT-fusion\cof062.txt

Dataset:

Processing Primitives Log

Processing • Frame: 4638 Entries in queue: 0 model Threads: 14 Active: 1 Finished: 4 Dead: 9

Models: Loading-Operation-FWD Last primitive: aircraft-leaves-arrival-zone_736 (finish) Current time: Sun Dec 13 18:02:34 CET 2009

Jess Visualisation Recognition

Thread: 49 • Model: Loading-Operation-FWD Rating: -233.65 Clutter: 50

Loading-Operation-FWD (249)
start: [15:58:52; 16:03:52]
finish: 16:03:53

Transporter-Positioning-FWD
start: [15:58:52; 16:03:52]
finish: [15:59:52; 16:03:52]

Transporter-Positioned-FWD
start: 16:01:52
finish: 16:03:47

Transporter-Leaves-FWD
start: 16:03:48
finish: 16:03:53

vehicle-positioned_185
start: 16:01:52
finish: 16:03:47
has-agent: transporter_127
has-location: right-fwd-ts-zone 29

vehicle-leaves-zone_237
start: 16:03:48
finish: 16:03:53
has-agent: transporter_127
has-location: right-fwd-ts-zone 29

Other Applications by Exchange of Ontology

Recognising Smart Home activities in the CASAS domain

Activities

Preparing dinner
Preparing lunch
Preparing breakfast
Preparing a snack
Preparing a beverage
Taking medication
Washing dishes
Listening to music
Watching TV
Bathing
Dressing
Grooming
Toileting
Doing laundry
Cleaning
Going out



Conclusions

- **SCENIOR meets essential generic requirements for real-time scene interpretation:**
 - Knowledge base in standardised language (here OWL-DL)
 - Incremental interpretation, predictive power
 - Multiple parallel interpretation threads
 - Context-dependent preference measure
- **The interpretation system can be automatically generated from OWL specifications.**
- **Expressiveness of OWL sets limits:**
 - SWRL rules are unwieldy
 - Probabilities cannot be represented conveniently