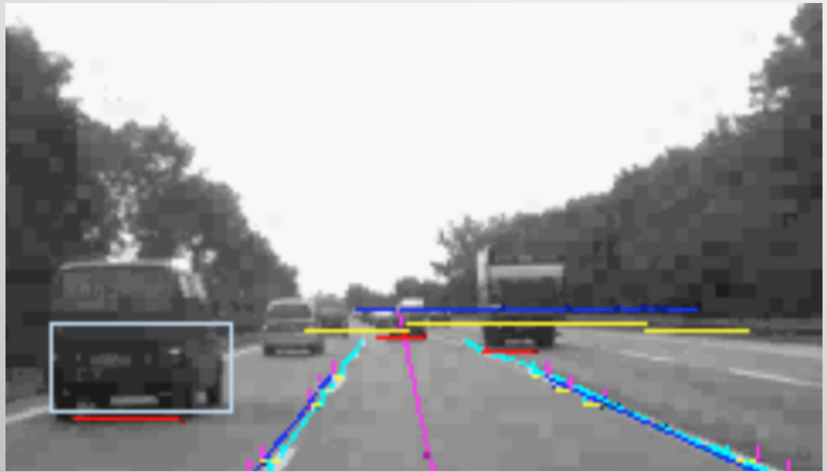
Ontology-based Realtime Activity Monitoring Using Beam Search

Wilfried Bohlken, Bernd Neumann
University of Hamburg

Lothar Hotz
HITeC

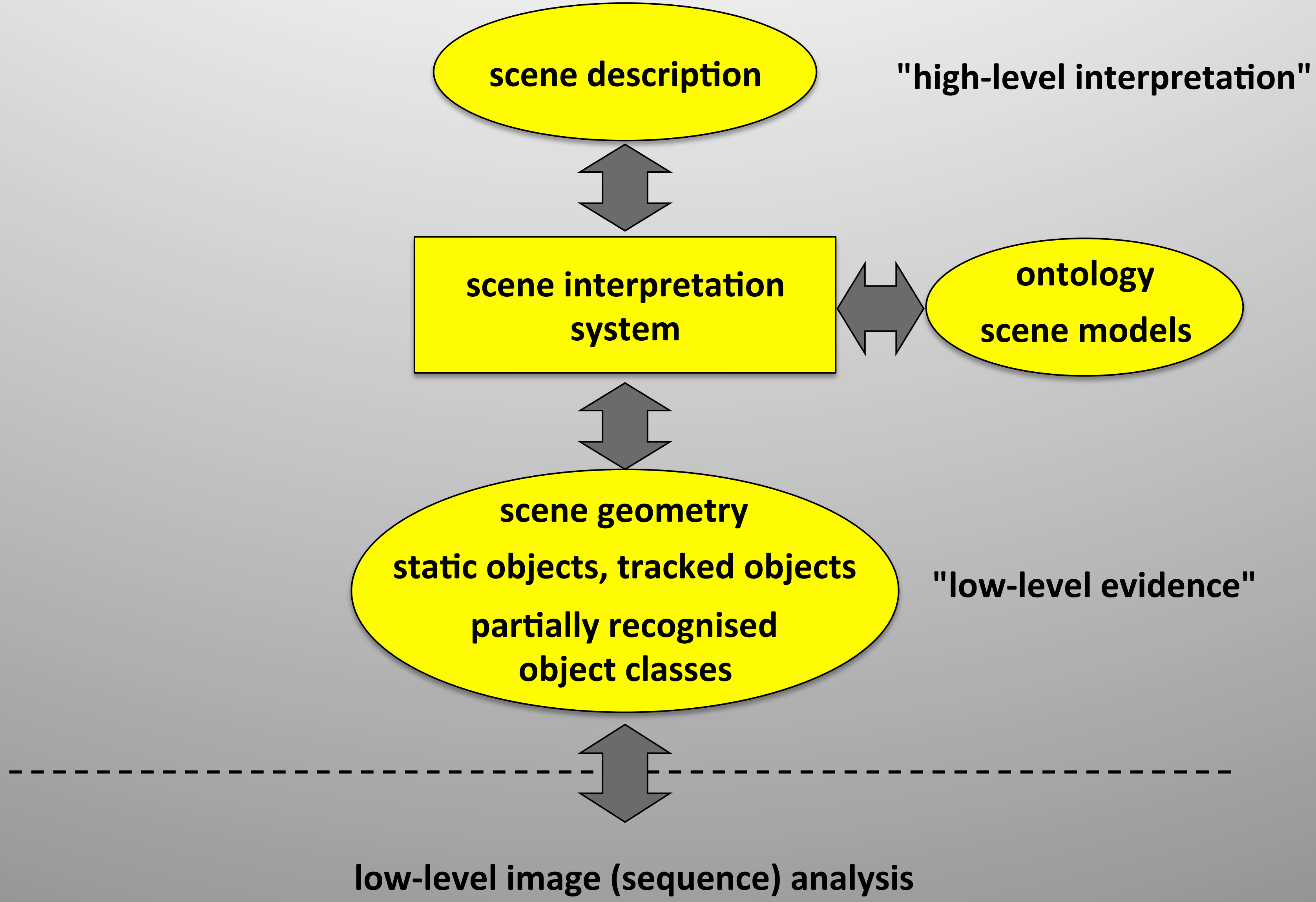
Patrick Koopmann
Cirquent GmbH

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

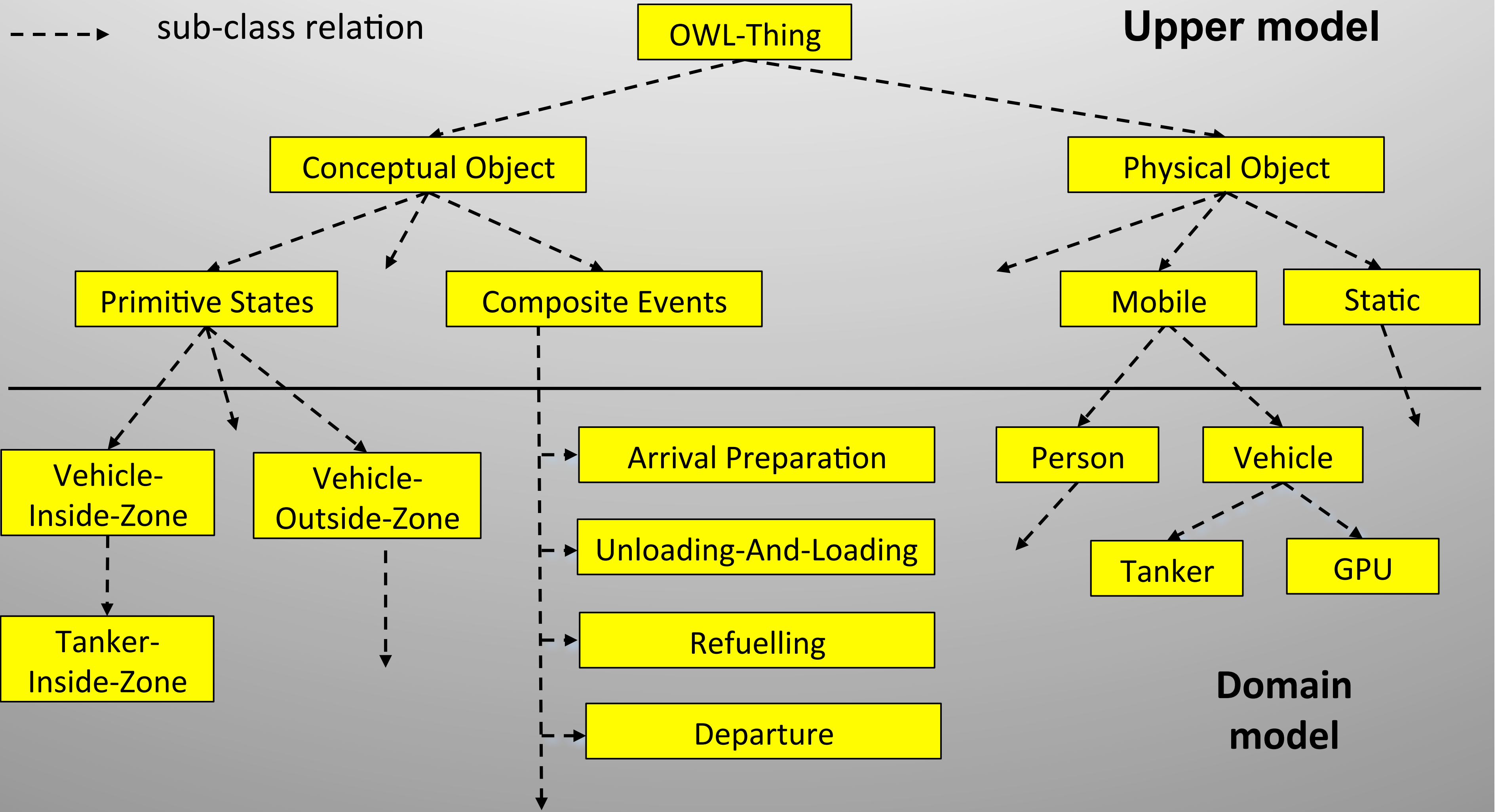
- Badler 1975 (conceptual descriptions of object motions)
- Neumann 1989 (natural language description of of traffic scenes)
- Rimey 1993 (Bayesian networks for vision control)
- Nagel 1999 (situation graph trees)
- Thonnat, Brémond 2007 (scenario recognition)
- Zhu & Mumford 2007 (stochastic grammar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

=> recognising hierarchical compositional structures

Representing Activity Concepts in OWL-DL

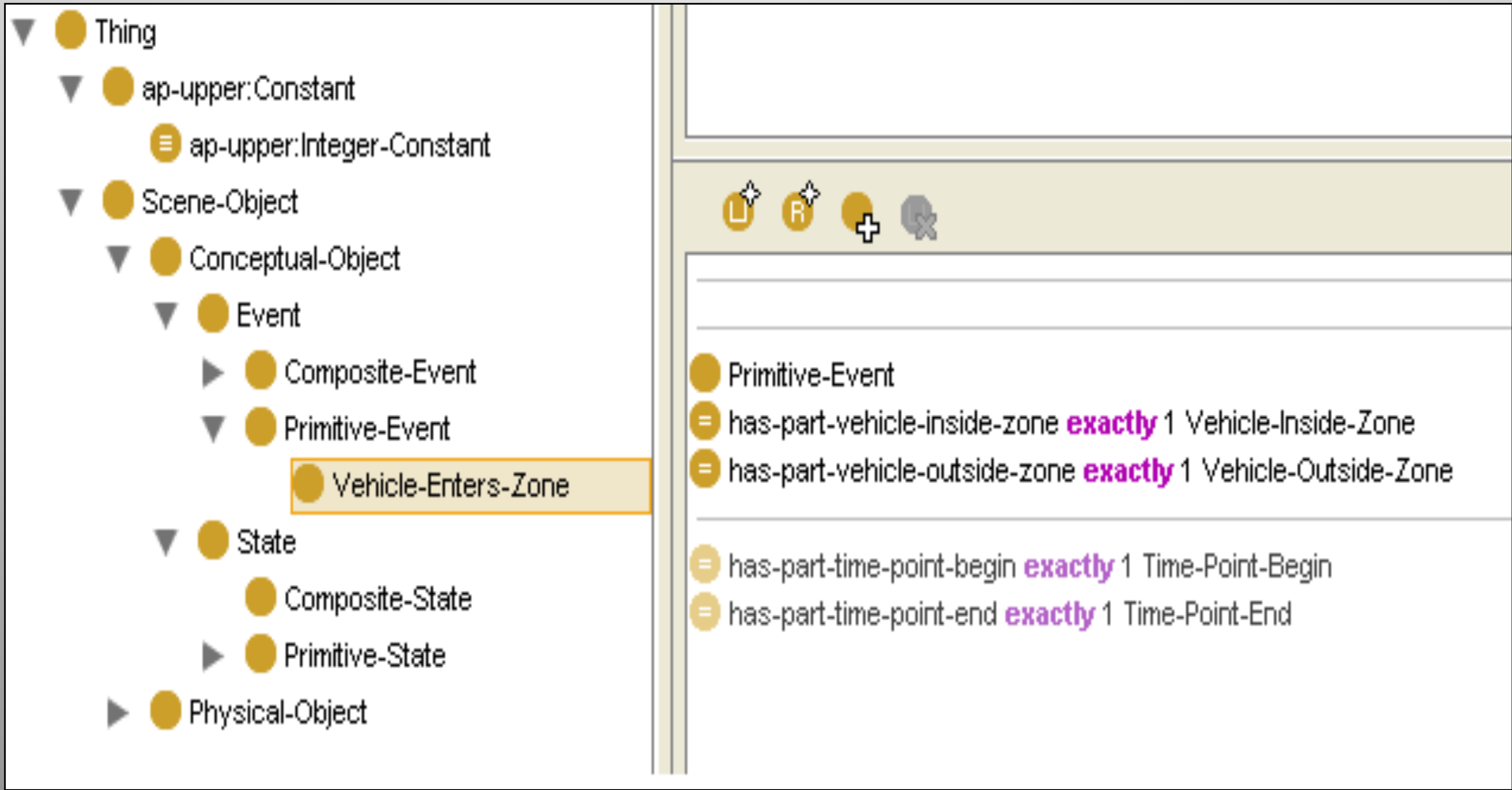
- **OWL is a standardised ontology language**
  - Definition of properties, aggregate taxonomies and partonomies
  - Knowledge editor Protégé in wide use
- **Powerful Description Logic reasoners support OWL-DL**
  - Useful services for large high-level knowledge bases
  - No support for stepwise recognition
  - 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

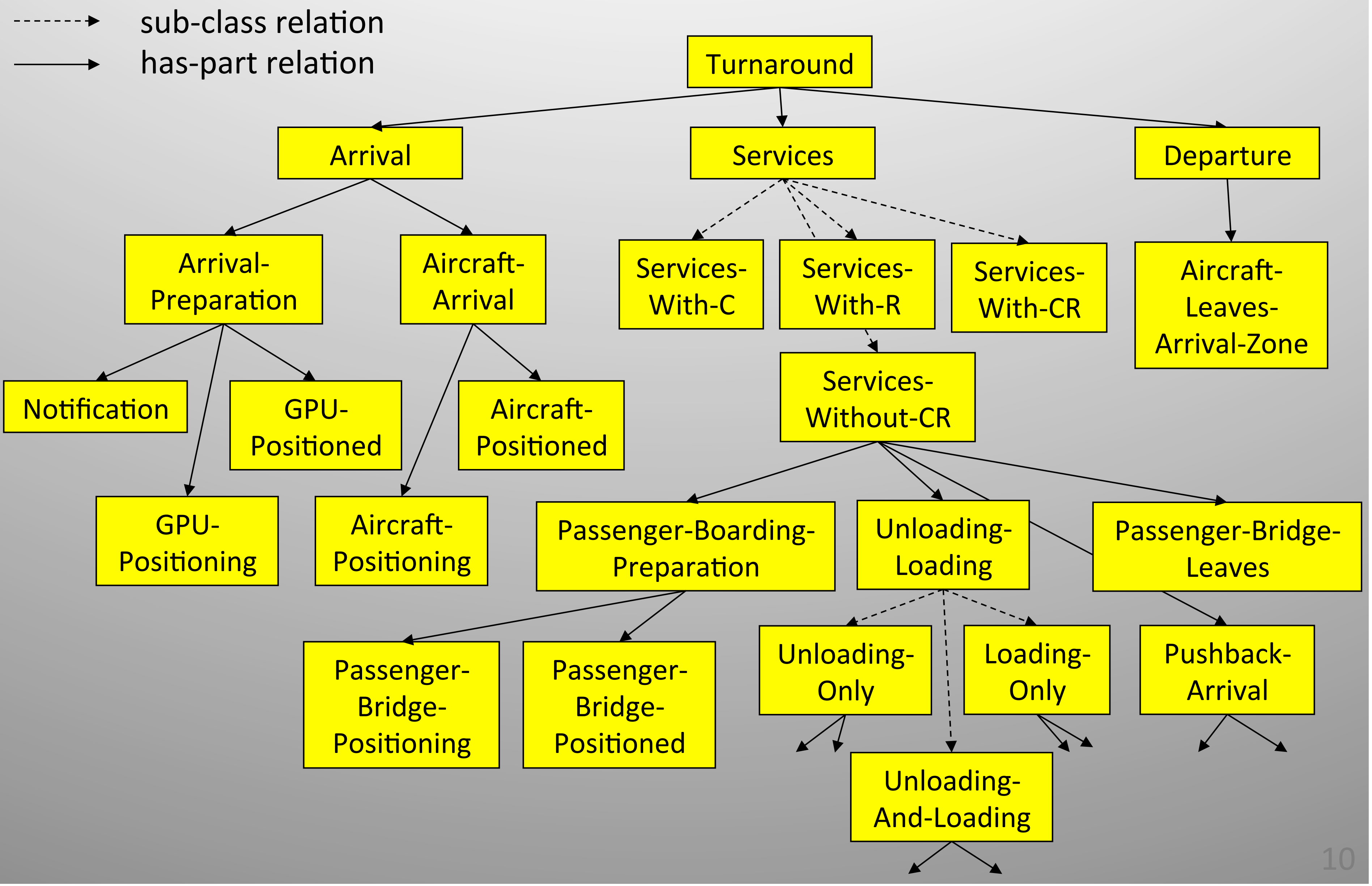


Using the Protégé Editor

Concepts are defined with taxonomical and binary relations (roles)



Compositional Hierarchy



Temporal Constraints in OWL

Monitoring service activities requires <u>quantitative</u> 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

Visit ⊑ Composite-Event ⊓
has-part1 exactly 1 Vehicle-Enters-Zone ⊓
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-Zove have the same agent and zone, respectively."

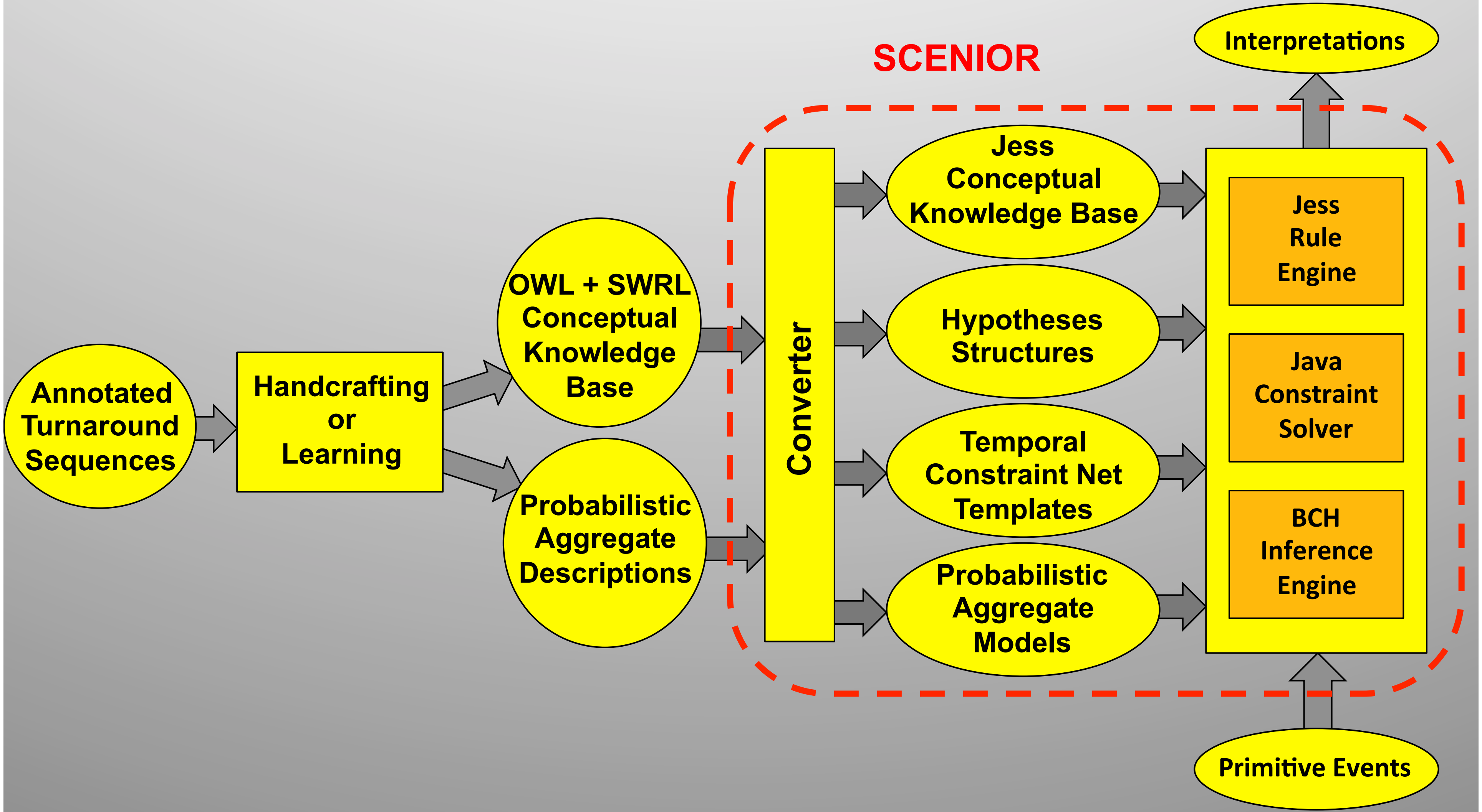
SWRL rule premise establishes variable names

Visit(?vis)
∧ has-part1(?vis, ?veh-enters)
∧ has-part2(?vis, ?veh-leaves)
∧ has-start-time(?vis, ?vis-st)
∧ has-finish-time(?vis, ?vis-ft)
∧ has-time-point(?veh-enters, ?veh-enters-tp)
∧ has-agent(?veh-enters, ?veh-enters-ag)
∧ has-zone(?veh-enters, ?veh-enters-zn)
∧ has-time-point(?veh-leaves, ?veh-leaves-tp)
∧ has-agent(?veh-leaves, ?veh-leaves-ag)
∧ has-zone(?veh-leaves, ?veh-leaves-zn)
→

SWRL rule consequence specifies identity constraints and temporal constraints

?vis-st = ?veh-enters-tp
∧ ?vis-ft = ?veh-leaves-ft
∧ ?veh-enters-ag = ?veh-leaves-ag
∧ ?veh-enters-zn = ?veh-leaves-zn
∧ ?veh-enters-tp ≤ ?veh-leaves-tp

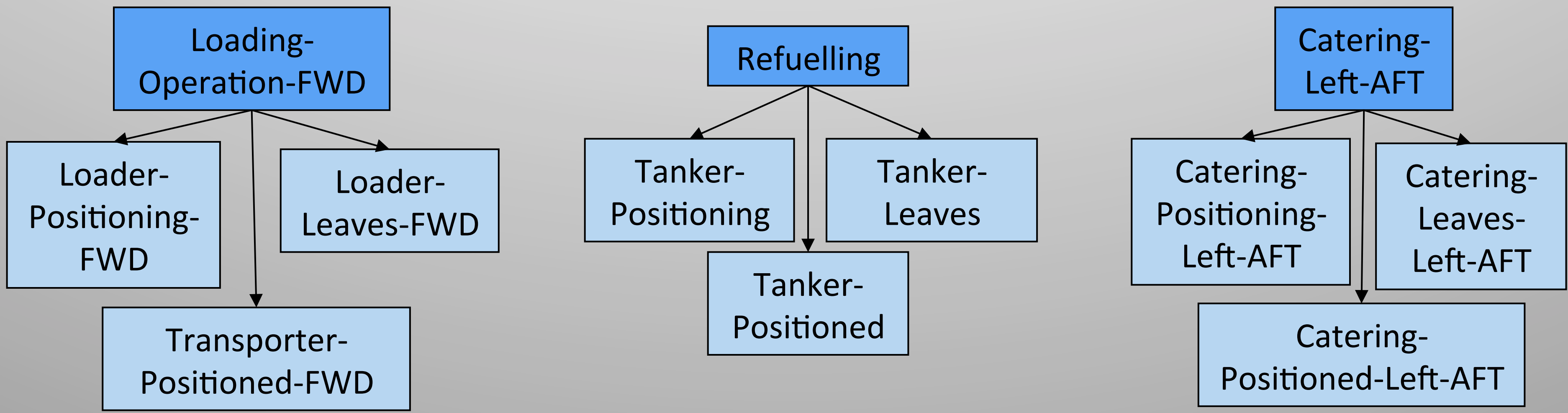
Transforming an OWL Knowledge Base into an Operational Interpretation System



Generating Hypotheses Structures for the JESS Working Memory (1)

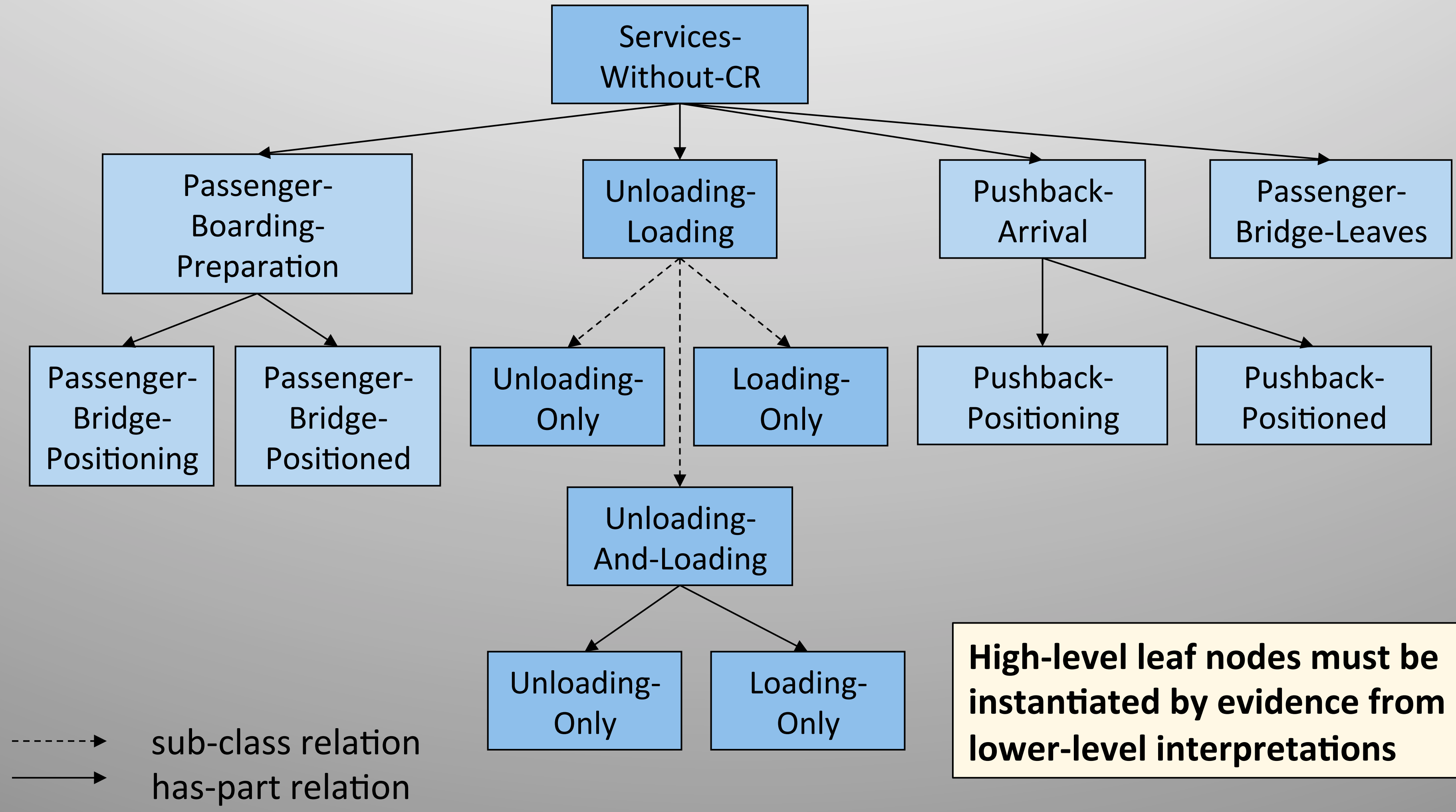
Hypotheses structures provide independent interpretation goals:

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



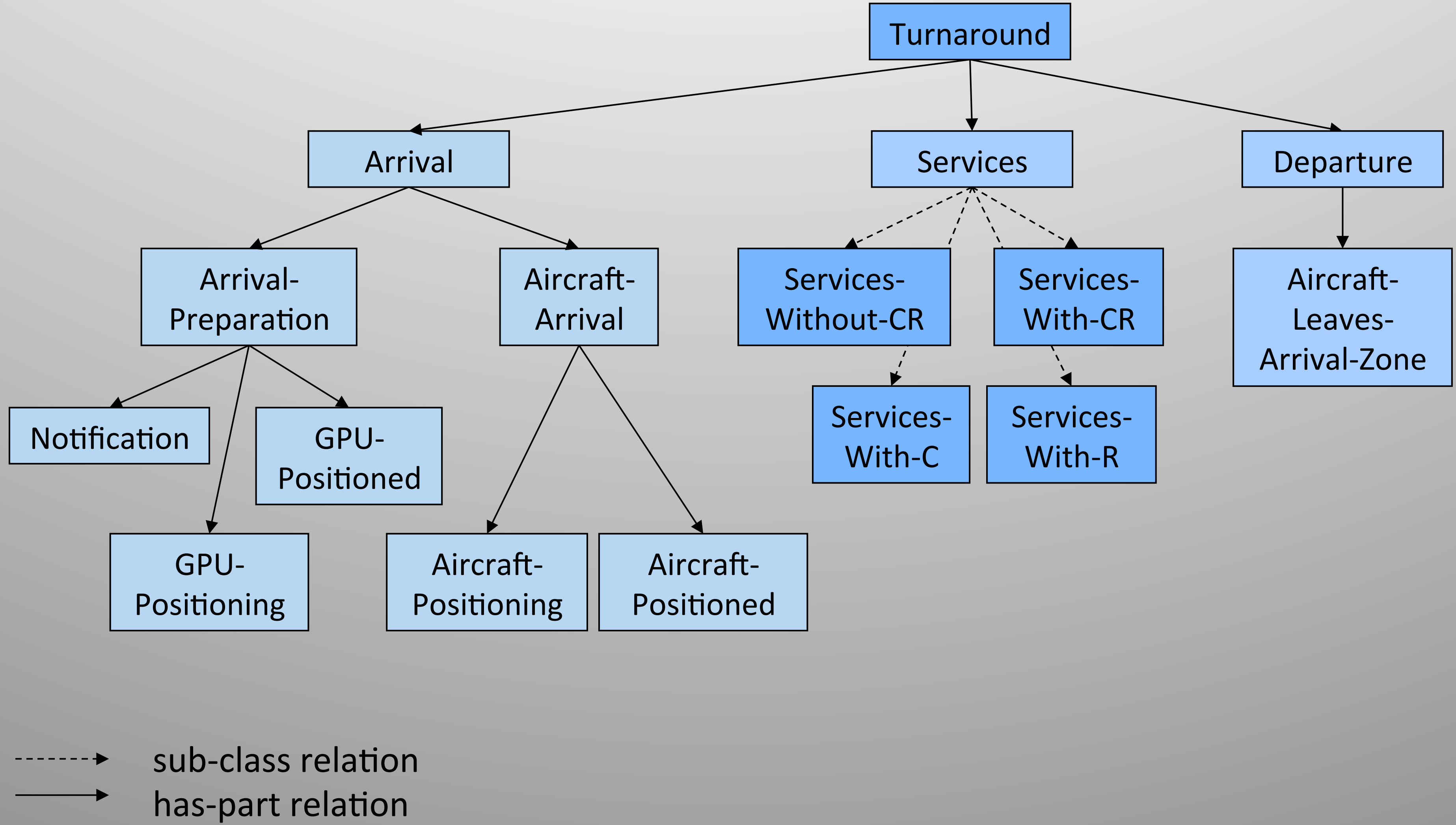
Low-level leaf nodes must be instantiated by evidence from low-level image analysis

Generating Hypotheses Structures for the JESS Working Memory (2)



High-level leaf nodes must be instantiated by evidence from lower-level interpretations

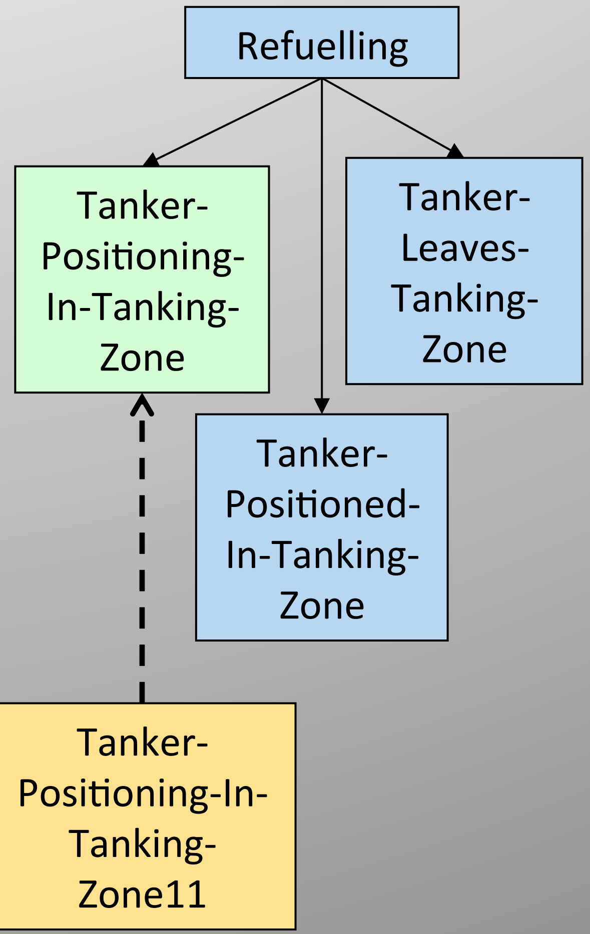
Generating Hypotheses Structures for the JESS Working Memory (3)



Generating JESS Interpretation Rules (1)

Evidence-assignment rule **for compositional leaf nodes**

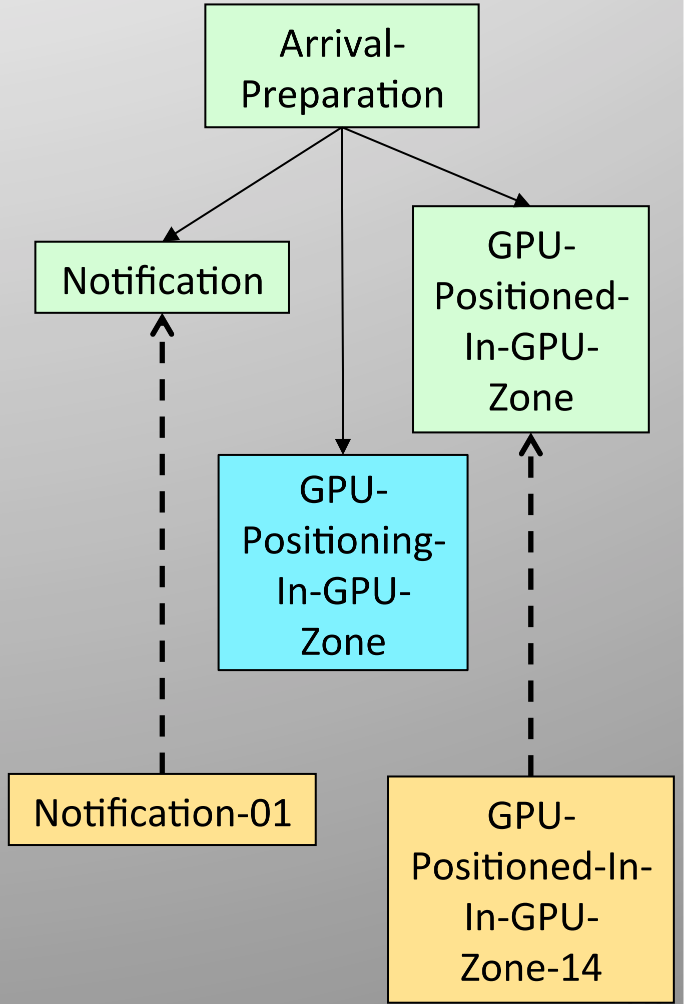
```
(defrule Refuelling_ea_rule
   ?e-id <- (Tanker-Positioning-In-Tanking-Zone
                       (name ?tpg_17)
                       (status evidence))
   ?h-id <-(Tanker-Positioning-In-Tanking-Zone
                       (name ?tpg_h)
                       (status ?status_1))
   (test (or (eq ?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**

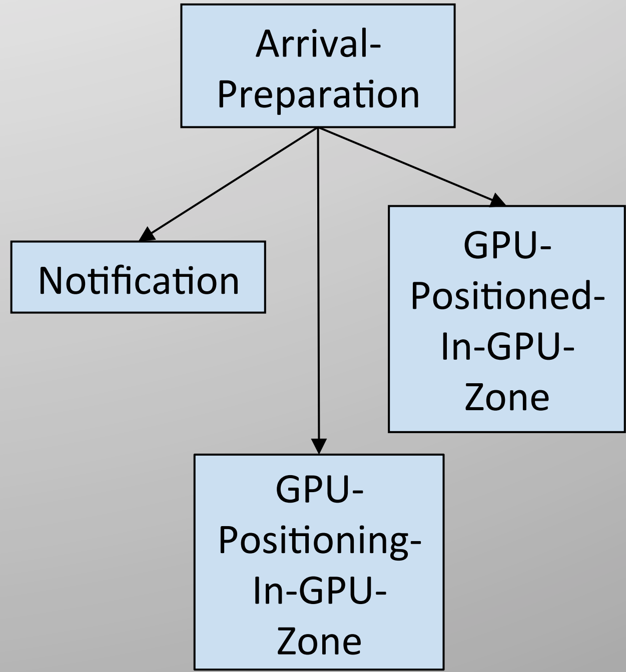
```
(defrule Arrival-preparation_ai_rule

    ?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)))
```

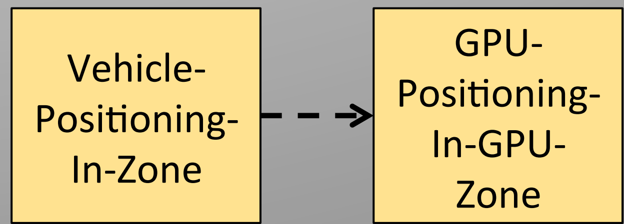


Generating JESS Interpretation Rules (3)

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

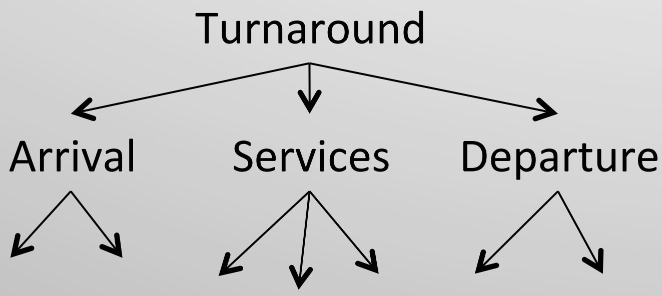
- Hypotheses structures are initialised as independent interpretation threads.
- New evidence is assigned to all matching threads or to clutter.
- Interpretation threads are cloned in case of multiple assignment possibilities.
- Low-rating threads exceeding the beam width are discarded.

Implementation in SCENIOR:

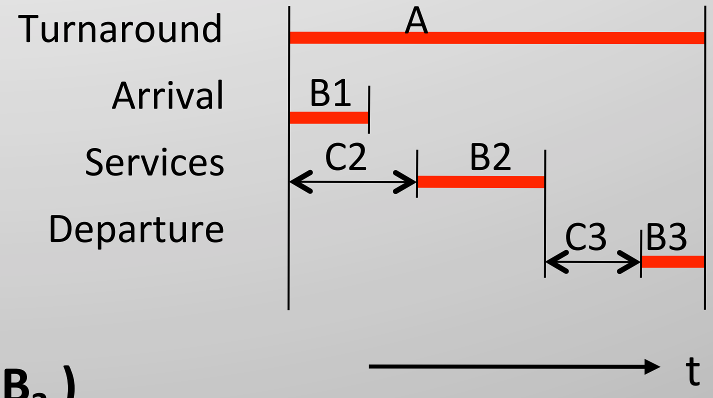
- SCENIOR can accommodate up to 100 threads.
- Ca. 800 threads are created for a typical turnaround scene.

Probabilistic Preference Measure Based on Aggregate JPDs

Aggregate partonomy



Temporal aggregate structure



Aggregate JPD $P_{Turnaround}(A\ B_1\ C_2\ B_2\ C_3\ B_3\ )$

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

For Bayesian Compositional Hierarchies (BCHs):

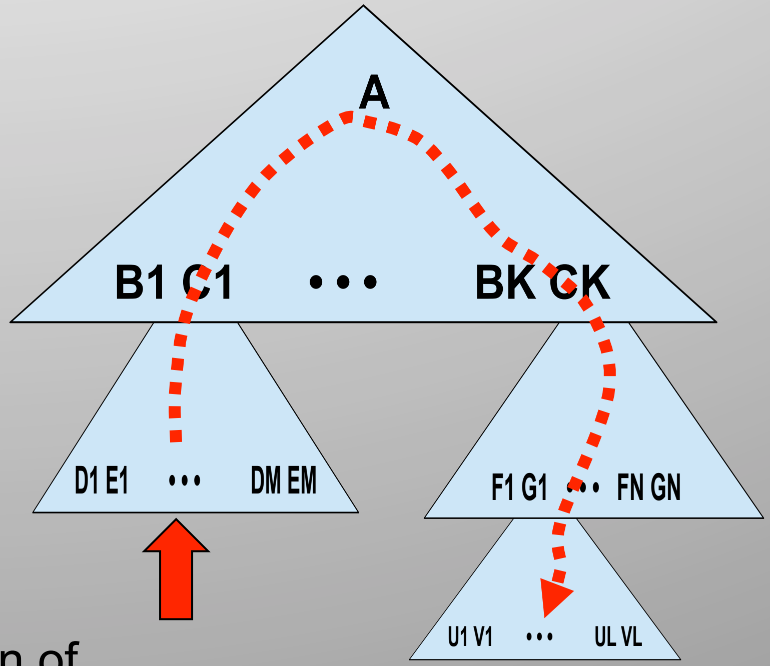
Scene JPD $P^{m}_{Scene} = p_m\ P'_{Turnaround}\ P'_{Arrival} \dots P'_{Refuelling} \dots P'_{Pushback}\ P_{clutter}$

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

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**

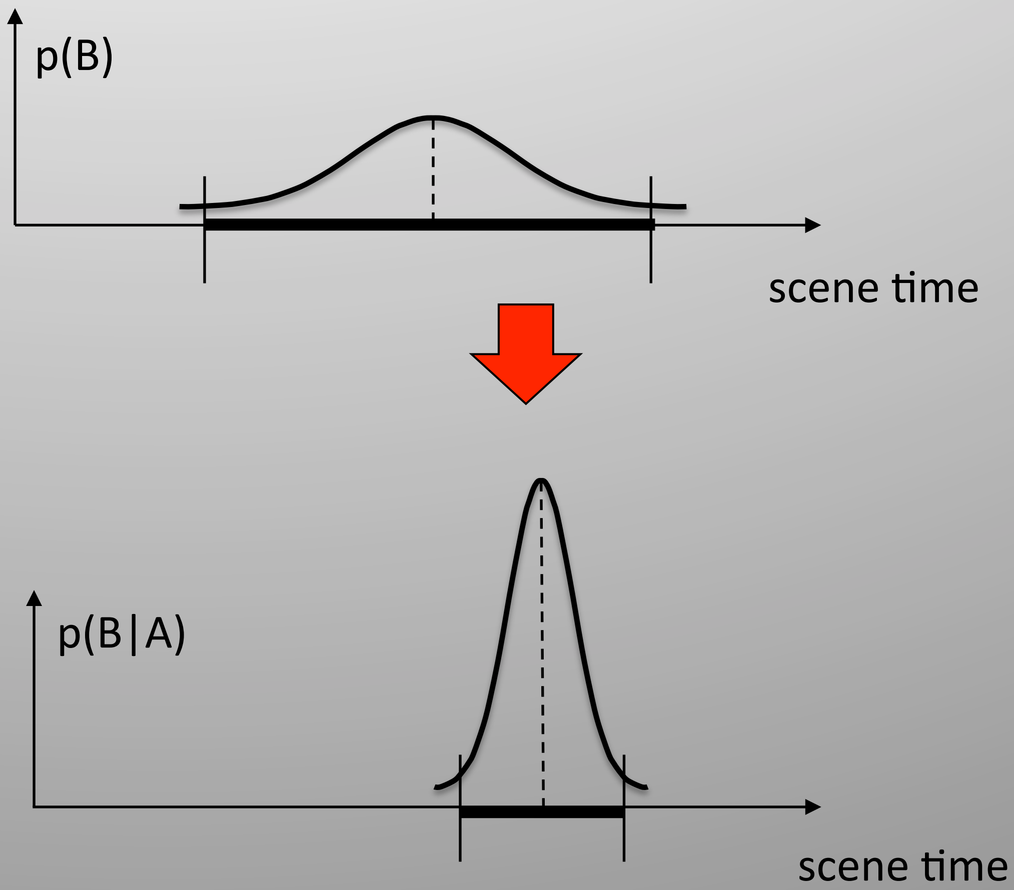
➡ **obtain dynamic priors (context-dependent)**

begin of
`Unloading-Operation`
at t = 26

Dynamic Priors for Multivariate Gaussians

Gaussian multivariate distributions allow highly efficient probability propagation.

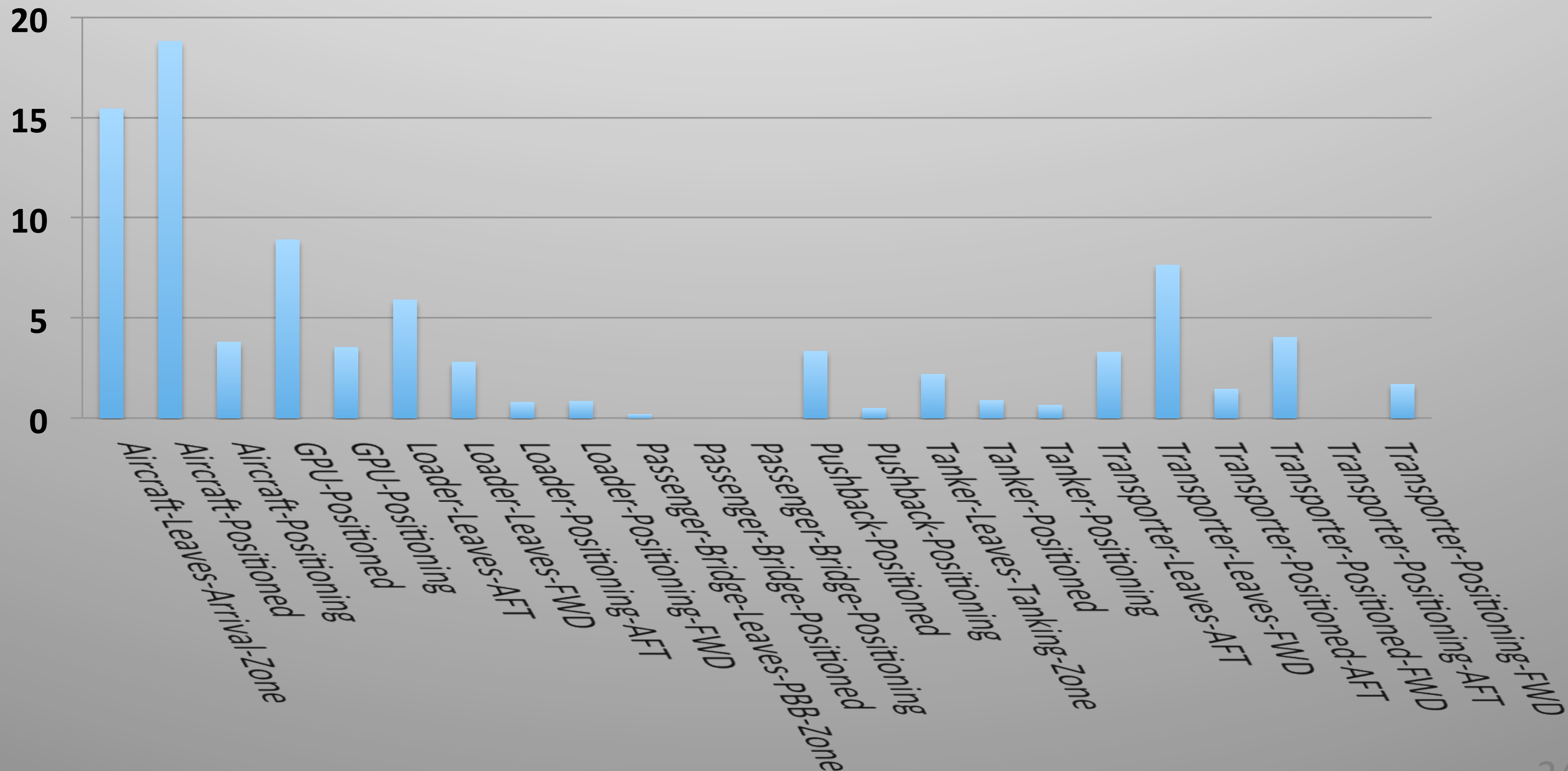
Expected temporal distribution of event B is changed by propagating observed time of event A



Experiments: False Positives

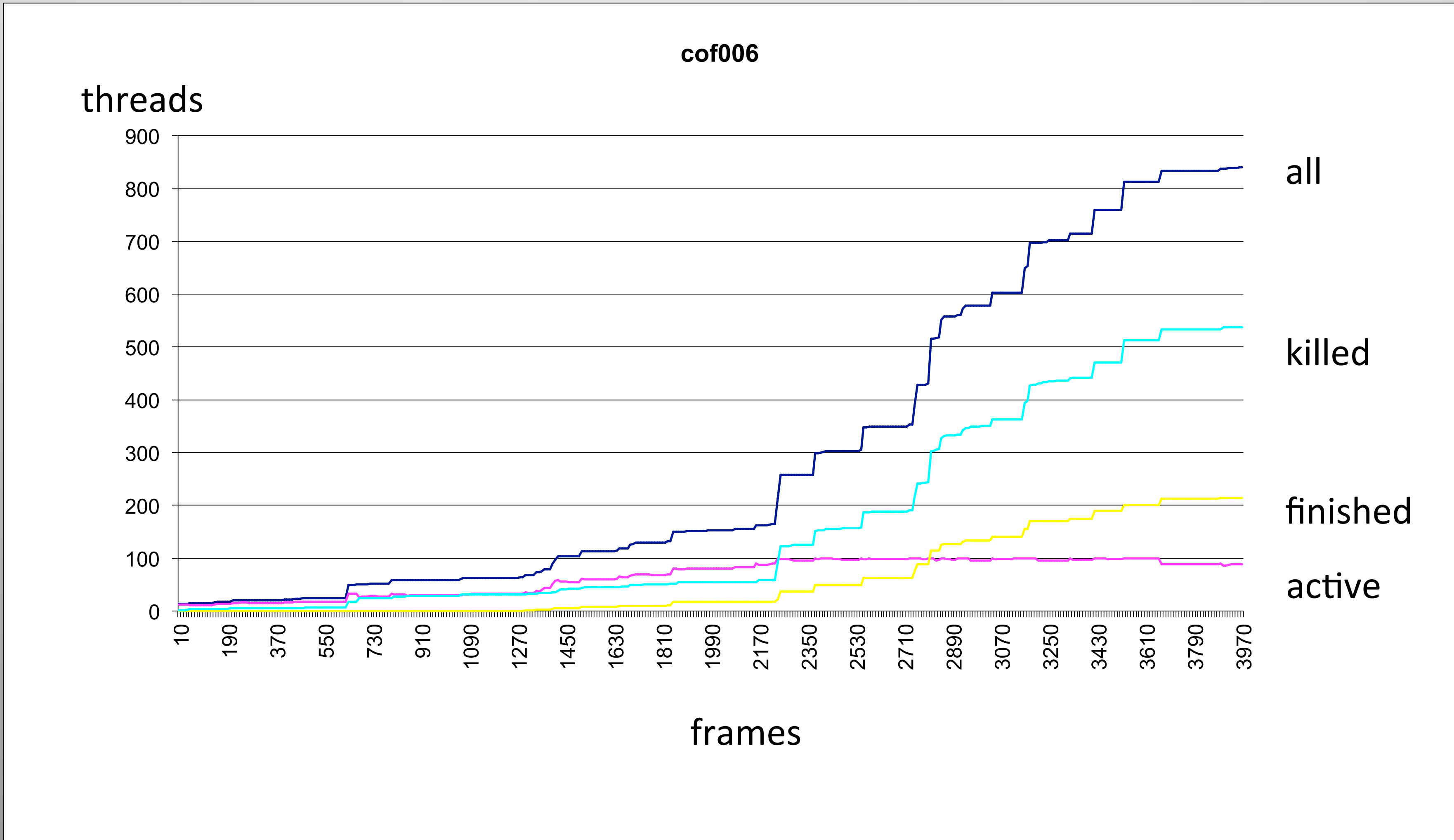
Low-level image analysis noise and unrelated scene activities have caused a large number of false positives.

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).

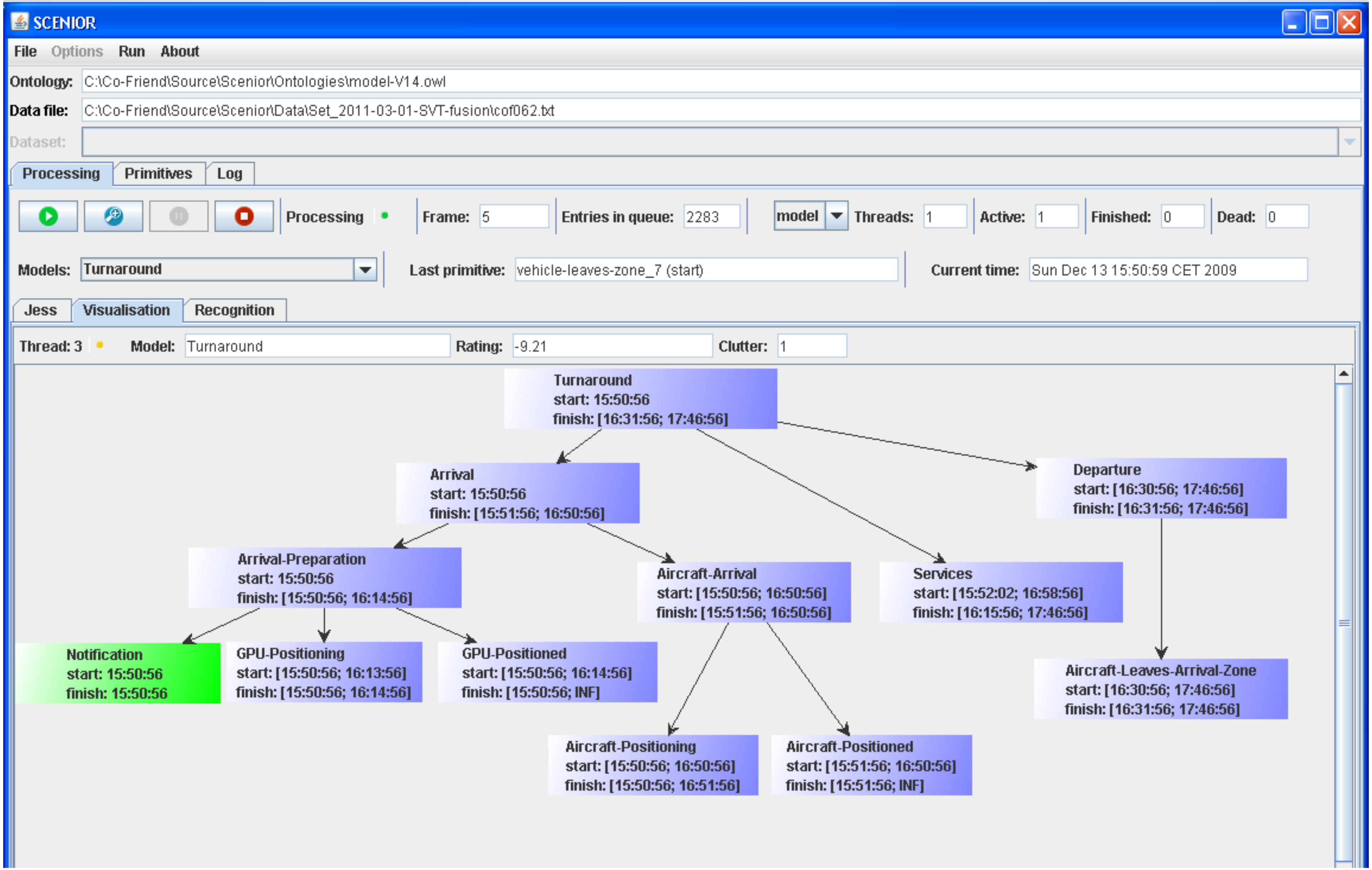


Recognition Results

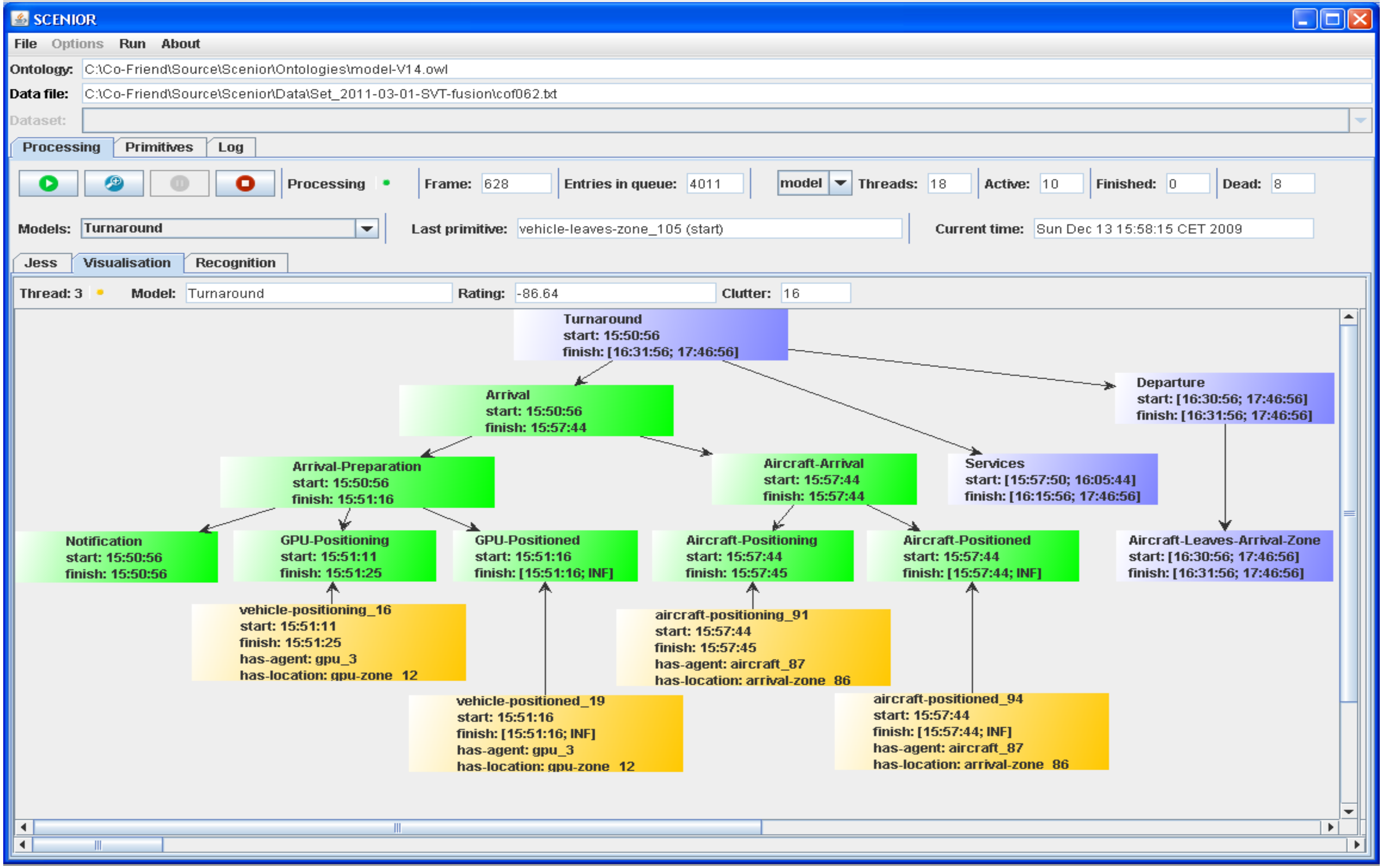
- Models trained on 32 annotated turnaround sequences, tested on 20 other sequences
- Complete turnarounds recognised for all but 3 highly irregular sequences
- 75% of all activities "correctly" recognised (overlap with annotated interval)

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

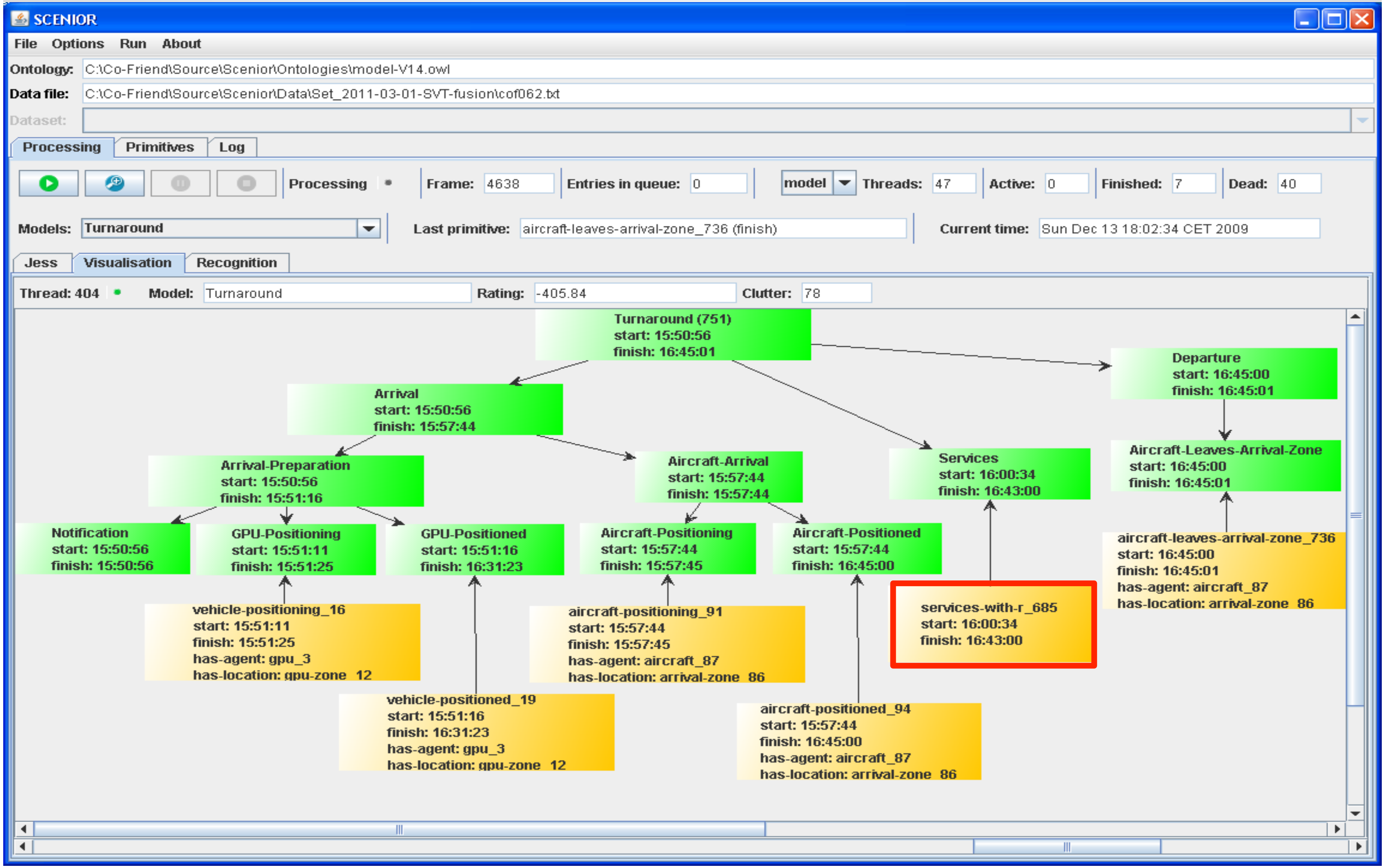
Turnaround Interpretation Log: Notification



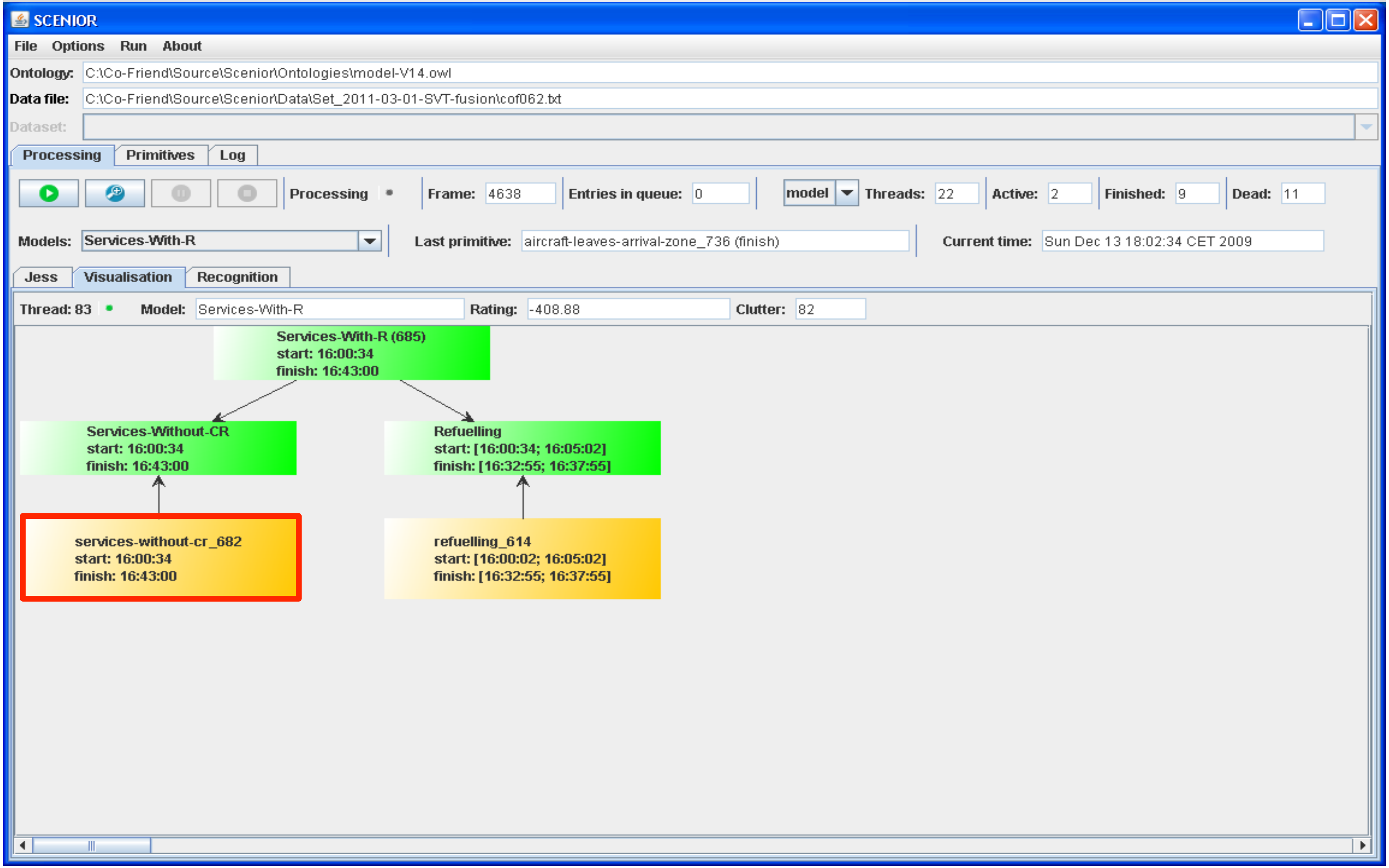
Turnaround Interpretation Log: Arrival



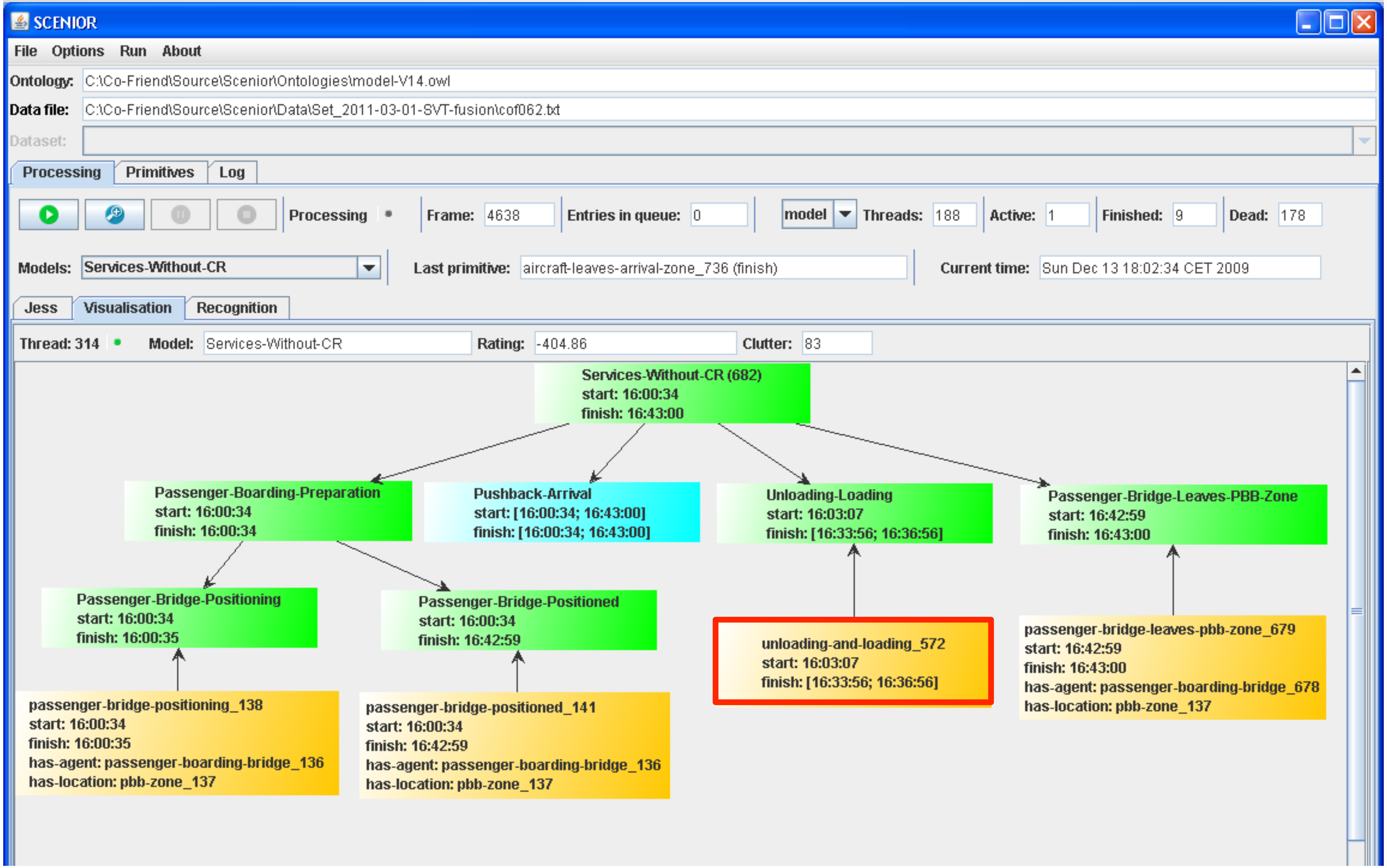
Turnaround Interpretation Log: Complete



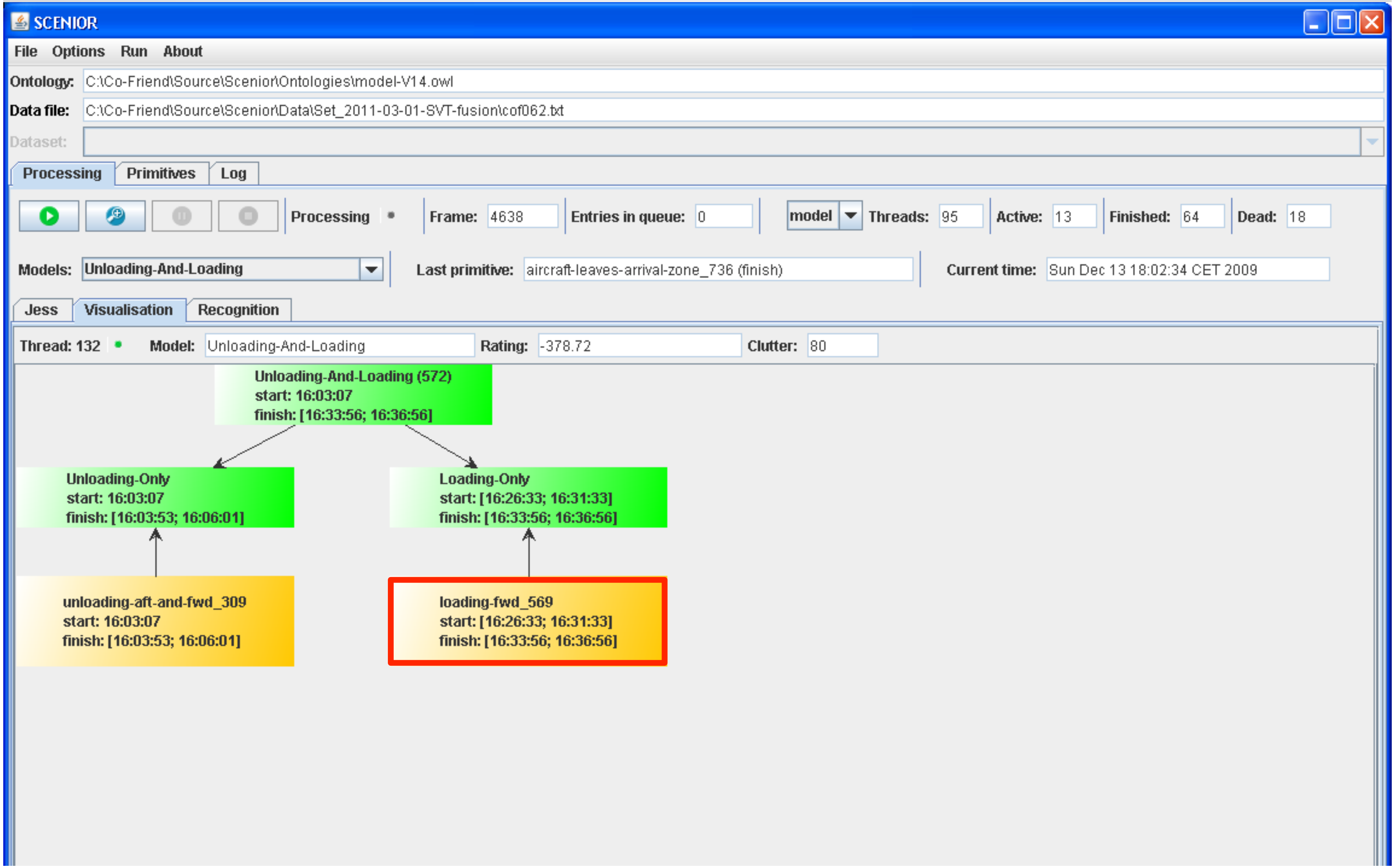
Turnaround Interpretation Log: Services with Refuelling



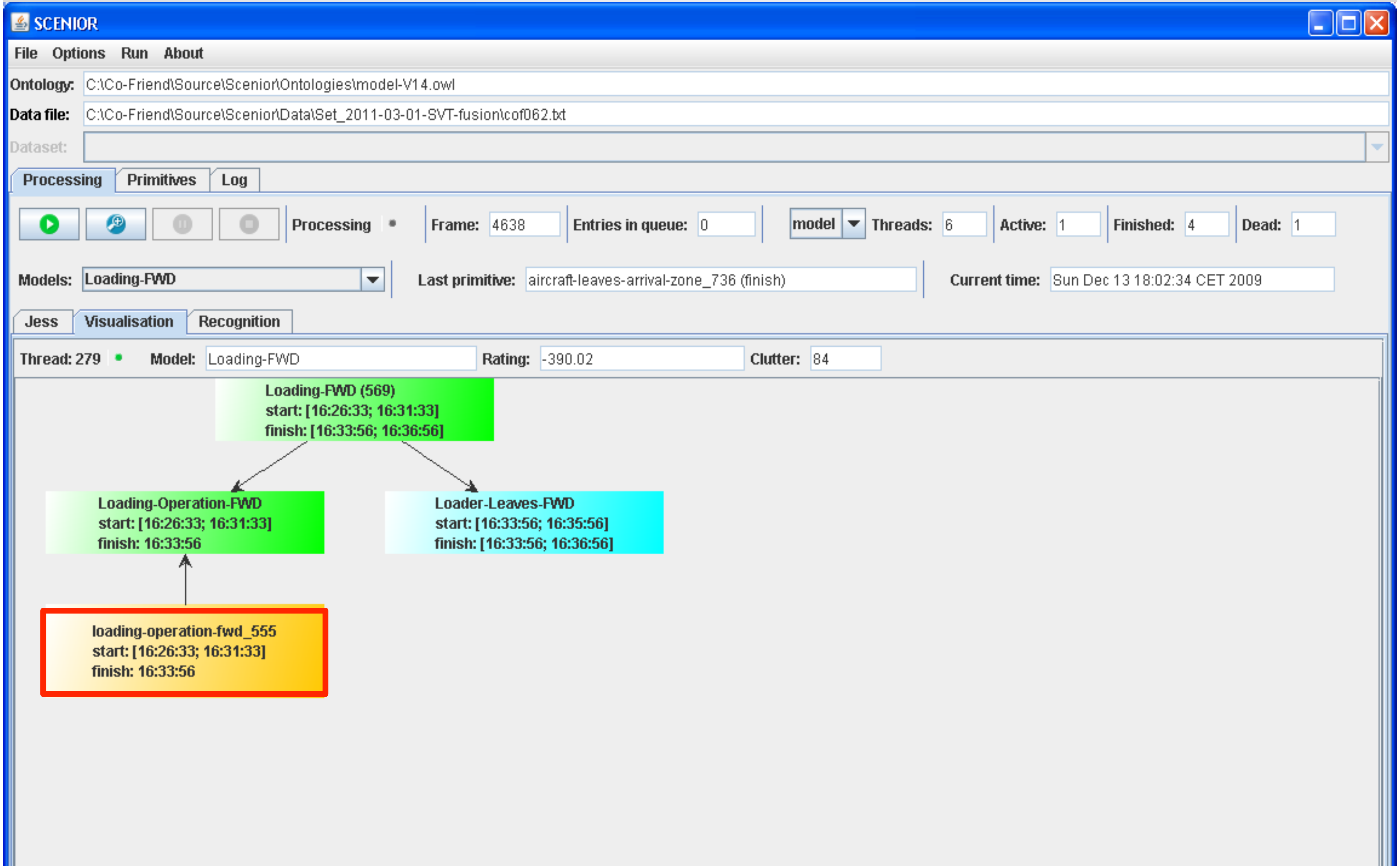
Turnaround Interpretation Log: Services without Catering and Refuelling



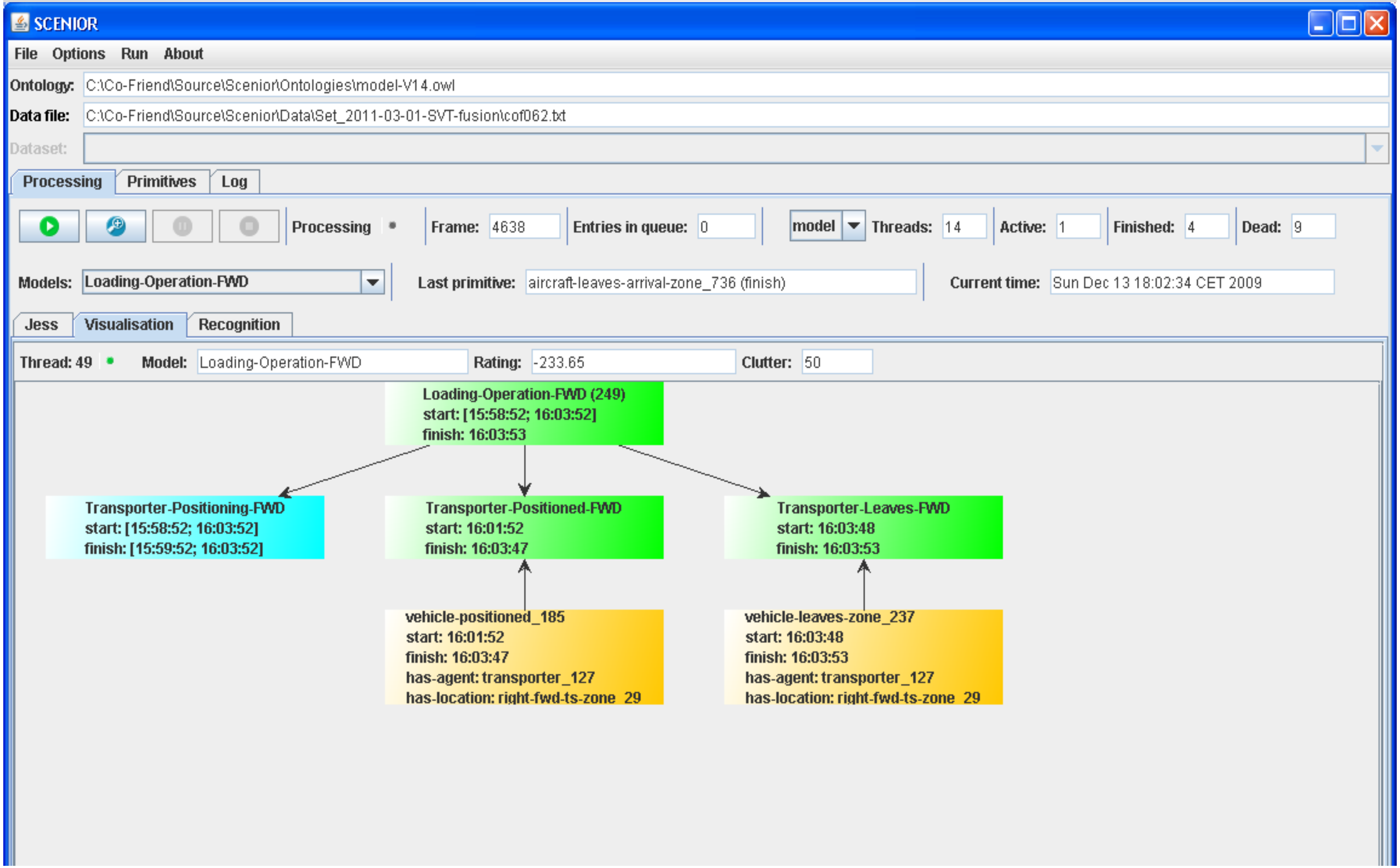
Turnaround Interpretation Log: Unloading and Loading



Turnaround Interpretation Log: Loading Forward



Turnaround Interpretation Log: Loading Operation Forward

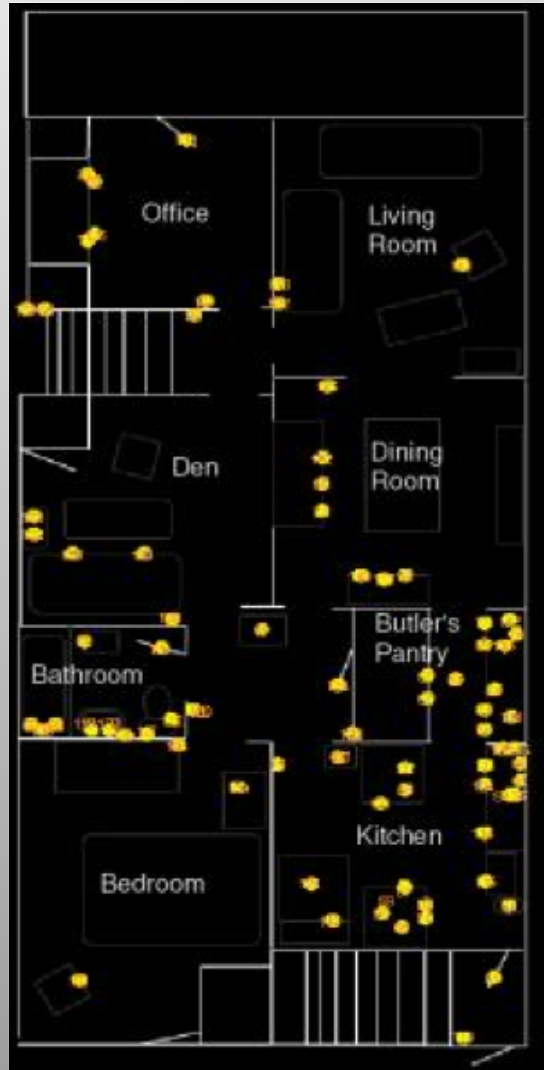


Other Applications by Exchange of Ontology

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