Linköpings universitet

AIICS Autonomous UAV Technologies

Stream Reasoning using Temporal Logic for Autonomous System

Fredrik Heintz, PhD

Knowledge Processing Laboratory/UASTech Laboratory

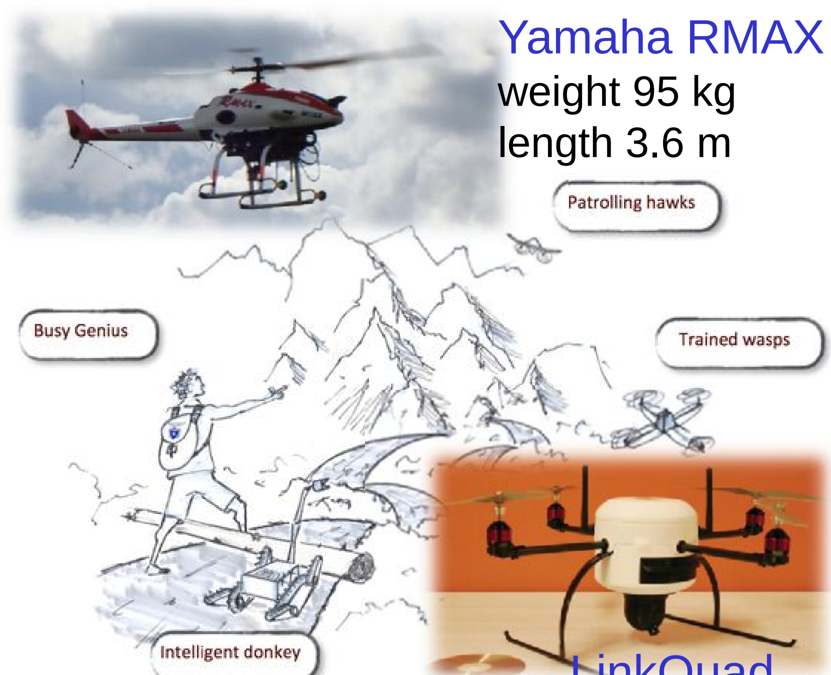
Artificial Intelligence and Integrated Computer Systems Division

Department of Computer and Information Science

Linköping University, Sweden

Collaborative Unmanned Aircraft Systems

A principled approach to building collaborative unmanned aircraft systems for complex missions.



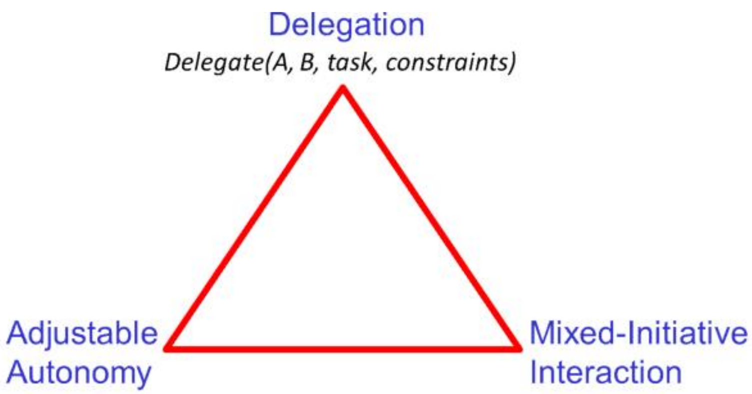
Yamaha RMAX
weight 95 kg
length 3.6 m

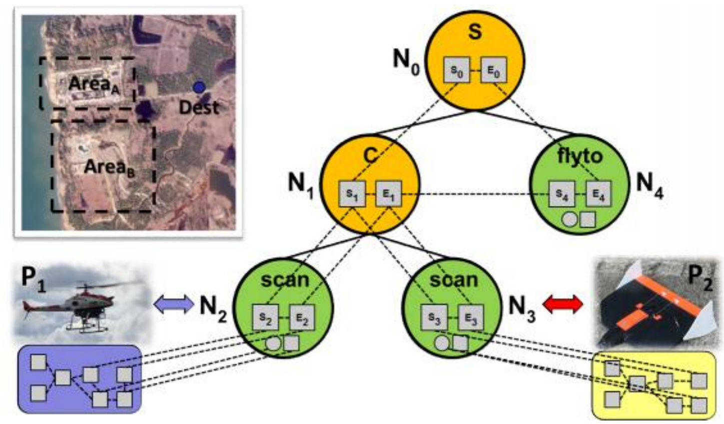
LinkQuad
weight ~1 kg
diameter ~70cm

SHERPA


SELECTED AUTONOMOUS FUNCTIONALITIES
LINKÖPING UNIVERSITY, SWEDEN
DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
AUTONOMOUS UNMANNED AERIAL VEHICLE TECHNOLOGIES LAB

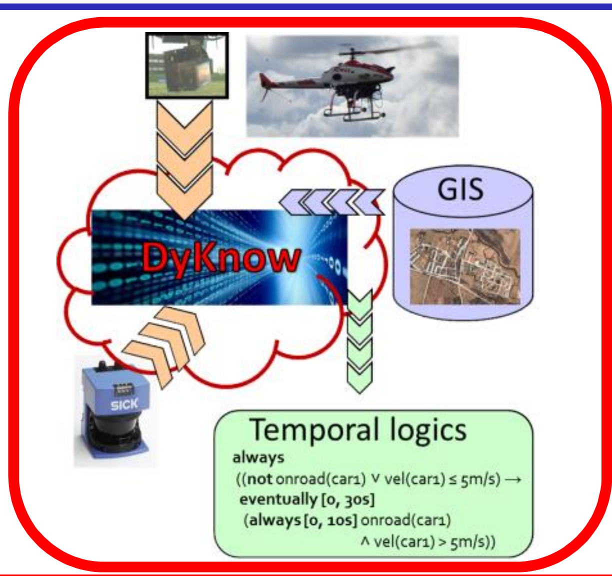
High-Level Research Overview



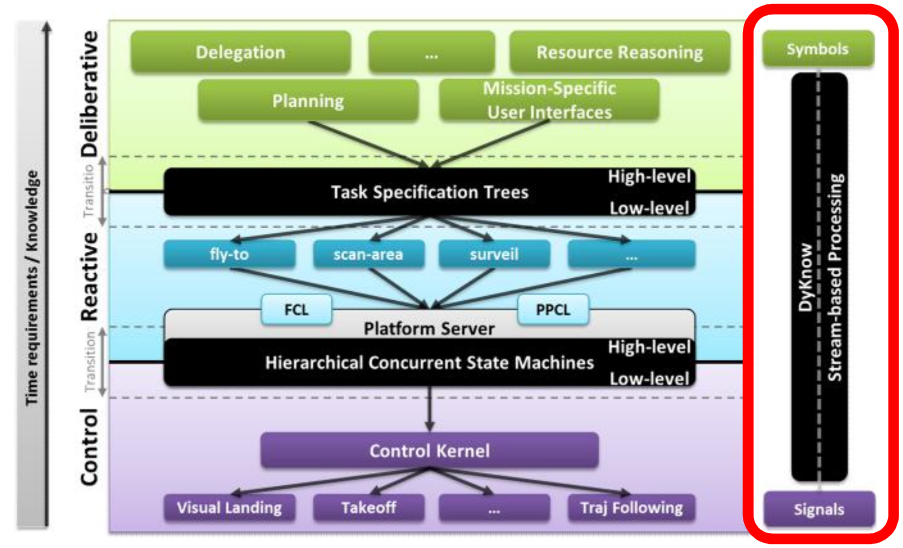


Delegation and task allocation through constraint satisfaction

Stream reasoning

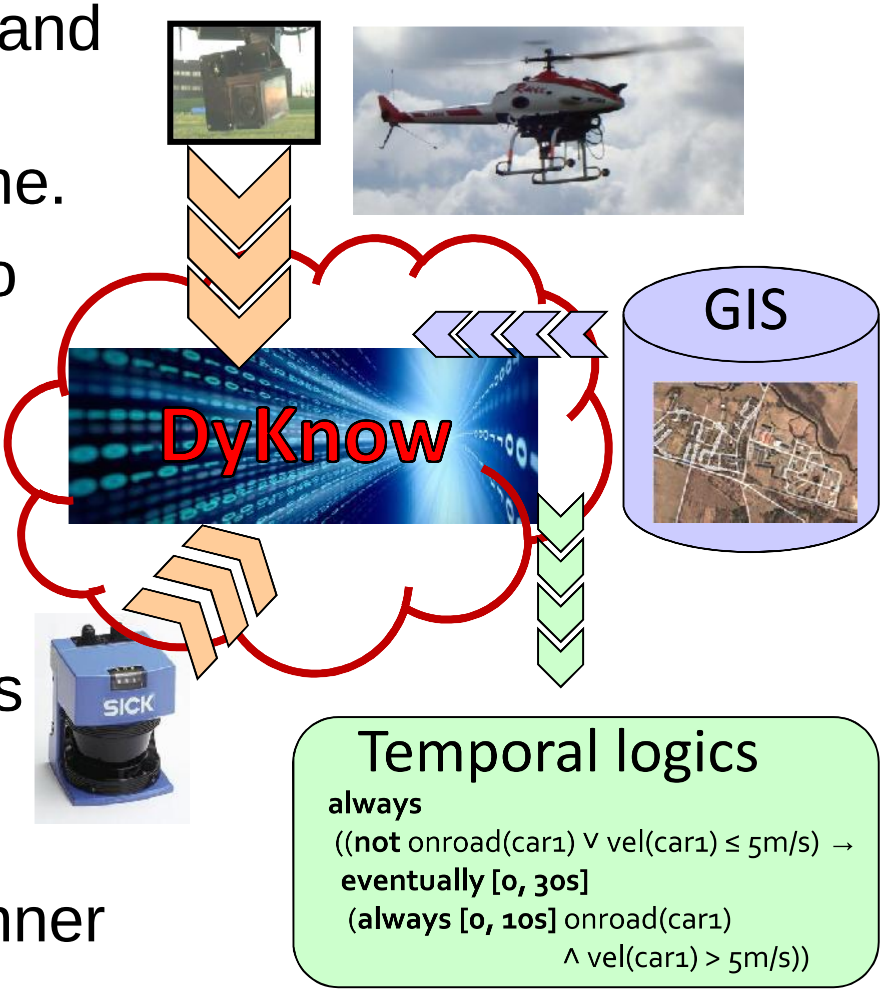


Architectures



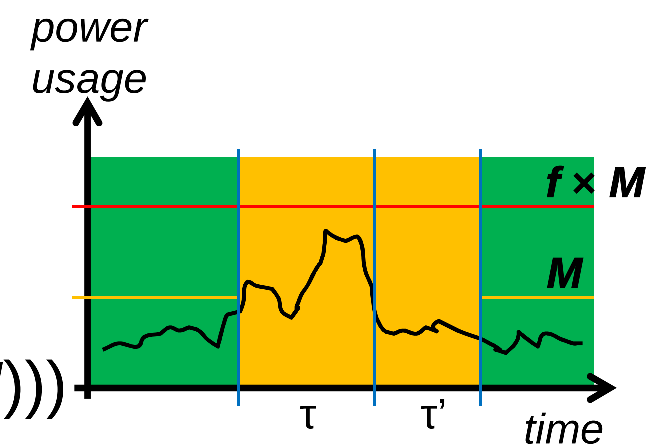
Stream Reasoning

- Autonomous systems produce and process sequences of values incrementally created at run-time.
- These sequences are natural to model as *streams*.
- *Stream reasoning* is incremental reasoning over streams.
- Stream reasoning approximates continuous reasoning with minimal latency necessary in order to react in a timely manner to changes in the environment.

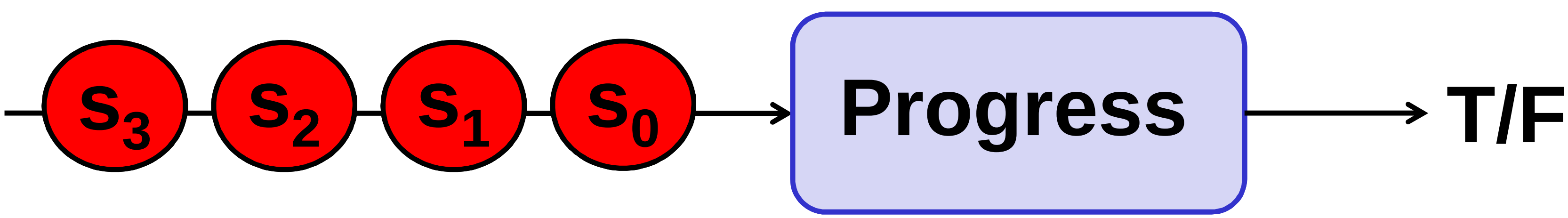


Stream Reasoning using Metric Temporal Logic

always $\forall$*uav*.((power_usage(*uav*) > *M*) →
((power_usage(*uav*) < *f* × *M*)
until[0, τ]
(**always**[0, τ’] power_usage(*uav*) ≤ *M*)))

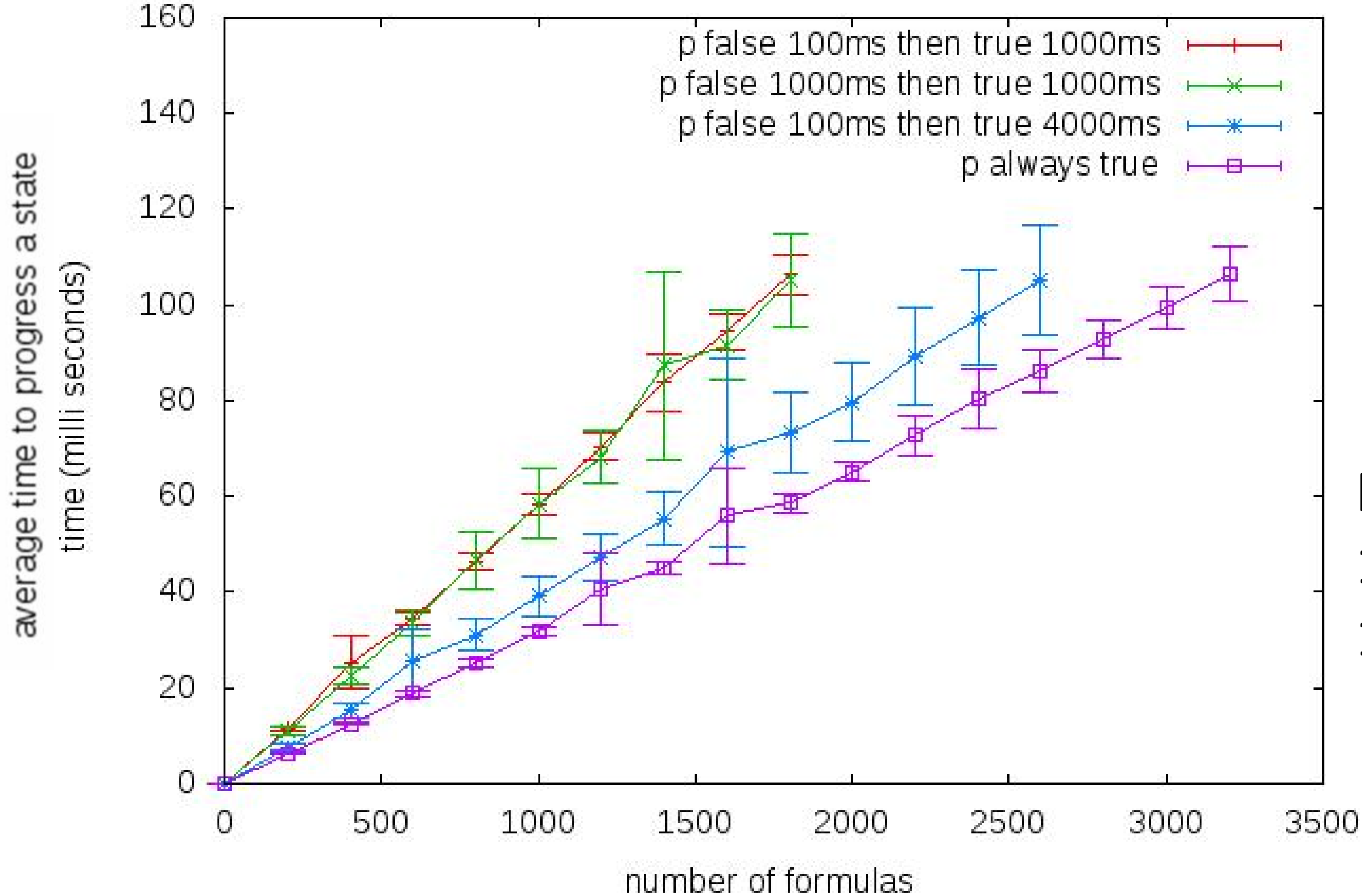


The semantics of these formulas is defined over infinite state sequences. Progression is one technique to check whether the current prefix is sufficient to determine the truth value of a formula.



Stream Reasoning using Metric Temporal Logic

always (¬p → **eventually** [0, 1000] **always** [0, 1000] p)



Pentium-M
1.4 GHz
1 GB RAM

Application: Execution Monitoring

If things can go wrong they probably will!

This implies the need for continual monitoring of an autonomous system and its environment in a principled, contextual, task specific manner which can be specified by the system itself!

always (eventually [0, t] (always [0,t'] speed(uav) < T))

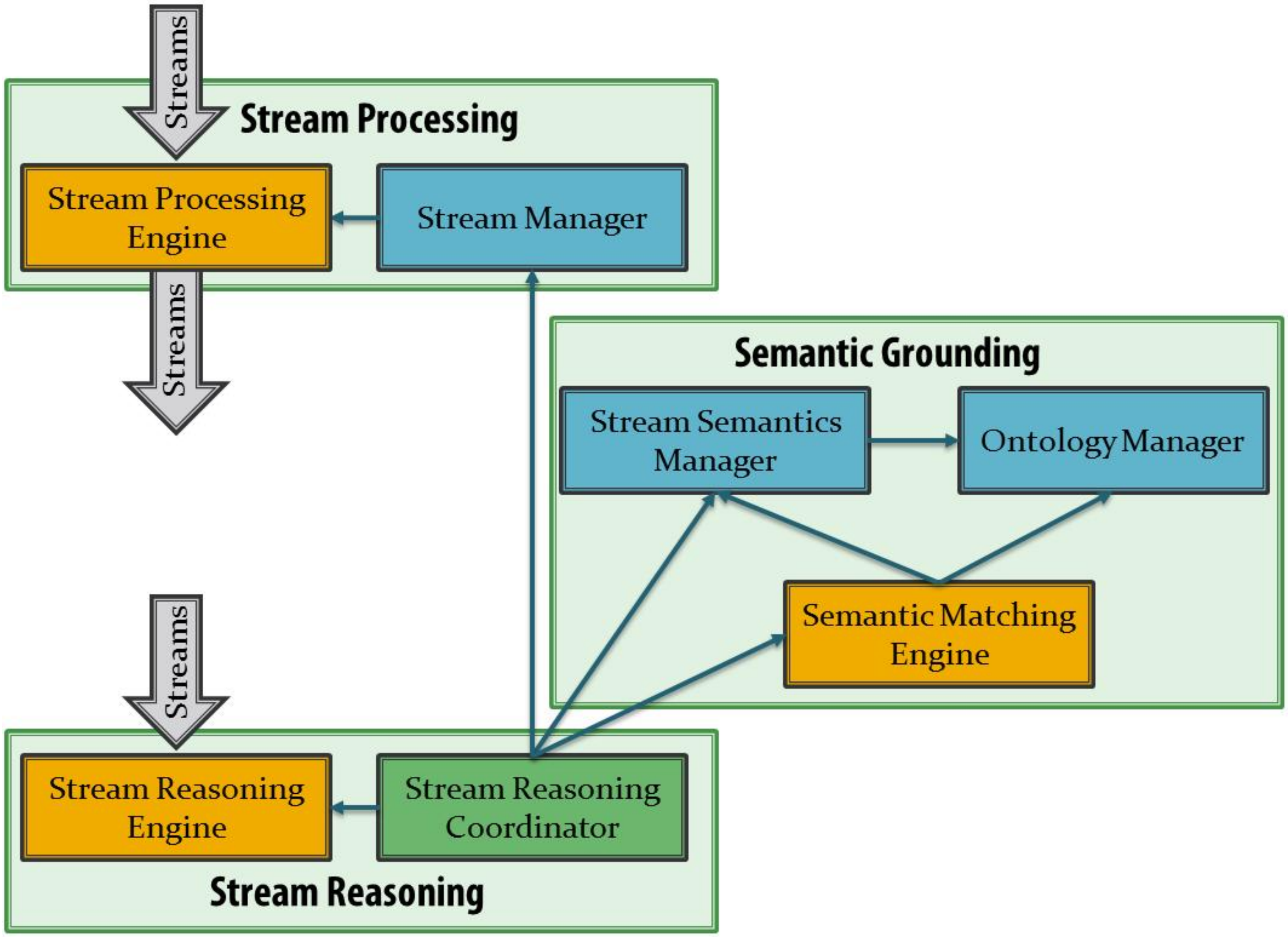
It should always be the case that within t time units from now, an interval of length t' should start where the UAV's speed stays below threshold T.

EXEC until [0, 5000] (¬EXEC ∧ altitude(uav) > 7)

The command should take less than 5 seconds to execute and when the execution is finished the altitude of the UAV should be above 7 meters.

DyKnow Semantically Grounded Stream Reasoning in ROS

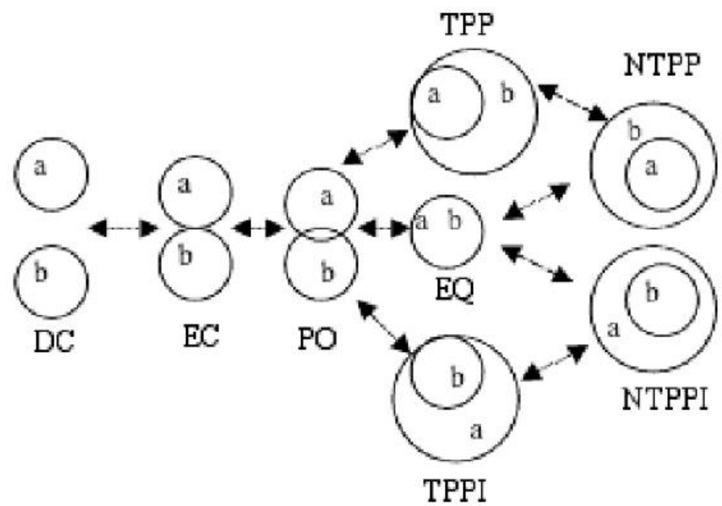
[Heintz IROS 2014]



Spatio-Temporal Stream Reasoning in DyKnow

[Heintz and de Leng ECAI 2014]

- The temporal reasoning is extended with spatial reasoning using for example RCC-8. RCC-8 defines 8 primitive relations and a composition table for qualitative constraint reasoning based on path consistency.
- Allows expressing conditions such as:
  - ∀*uav, restricted_area* **always** DC(*uav*, *restricted_area*)
  - ∀*uav, urban_area* **always** (PO(*uav*, *urban_area*) → **eventually** [0, 2min] altitude(*uav*) > 100m)



Spatio-Temporal Stream Reasoning in DyKnow

[Heintz and de Leng ECAI 2014]

- The temporal reasoning is extended with spatial reasoning using for example RCC-8. RCC-8 defines 8 primitive relations and a composition table for qualitative constraint reasoning based on path consistency.
- Allows expressing conditions such as:
  - ∀*uav, restricted_area* **always** DC(*uav*, *restricted_area*)
  - ∀*uav, urban_area* **always** (PO(*uav*, *urban_area*) → **eventually** [0, 2min] altitude(*uav*) > 100m)

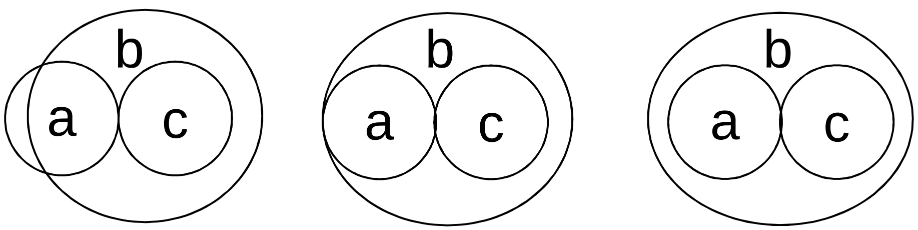
DC(urban_area1, restricted_area1)
DC(urban_area1, restricted_area2)
DC(urban_area1, urban_area2)
DC(urban_area1, road1)
DC(urban_area1, road2)
EC(road1, restricted_area1)
EC(road1, restricted_area2)
PO(road1, urban_area2)
DC(road1, road2)
PO(uav1, road2)
PO(uav1, urban_area2) …

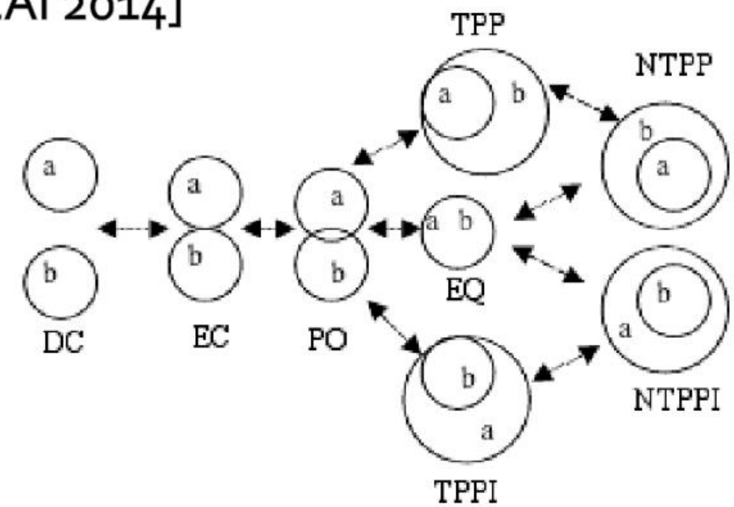
Spatio-Temporal Stream Reasoning in DyKnow

[Heintz and de Leng ECAI 2014]

Known: EC(a,c) $\wedge$ NTPP(c,b)

Deduced: PO(a,b) $\vee$ TPP(a,b)
$\vee$ NTPP(a,b)





DC(urban_area1, restricted_area1)
DC(urban_area1, restricted_area2)
DC(urban_area1, urban_area2)
DC(urban_area1, road1)
DC(urban_area1, road2)
EC(road1, restricted_area1)
EC(road1, restricted_area2)
PO(road1, urban_area2)
DC(road1, road2)
PO(uav1, road2)
PO(uav1, urban_area2) …

Spatio-Temporal Stream Reasoning in DyKnow

[Heintz and de Leng ECAI 2014]

- To handle incomplete spatial information we extend the first order logic to a three valued strong Kleene logic.

A and B	T	F	U
T	T	F	U
F	F	F	F
U	U	F	U

A or B	T	F	U
T	T	T	T
F	T	F	U
U	T	U	U

not A	
T	F
F	T
U	U

- The truth value of a spatial predicate P(a,b) given a set S of disjunctive base relations that hold between a and b is:
  - P(a,b) is true if $P \in S$ and $|S|=1$
  - P(a,b) is unknown if $P \in S$ and $|S|>1$
  - P(a,b) is false if $P \notin S$

always (PO(a,b) $\rightarrow$ **eventually** [0,2] DC(a,b))

always (PO(a,b) $\rightarrow$ **eventually** [0,2] DC(a,b))
$\wedge$ (U $\vee$ **eventually** [0,2] DC(a,b))

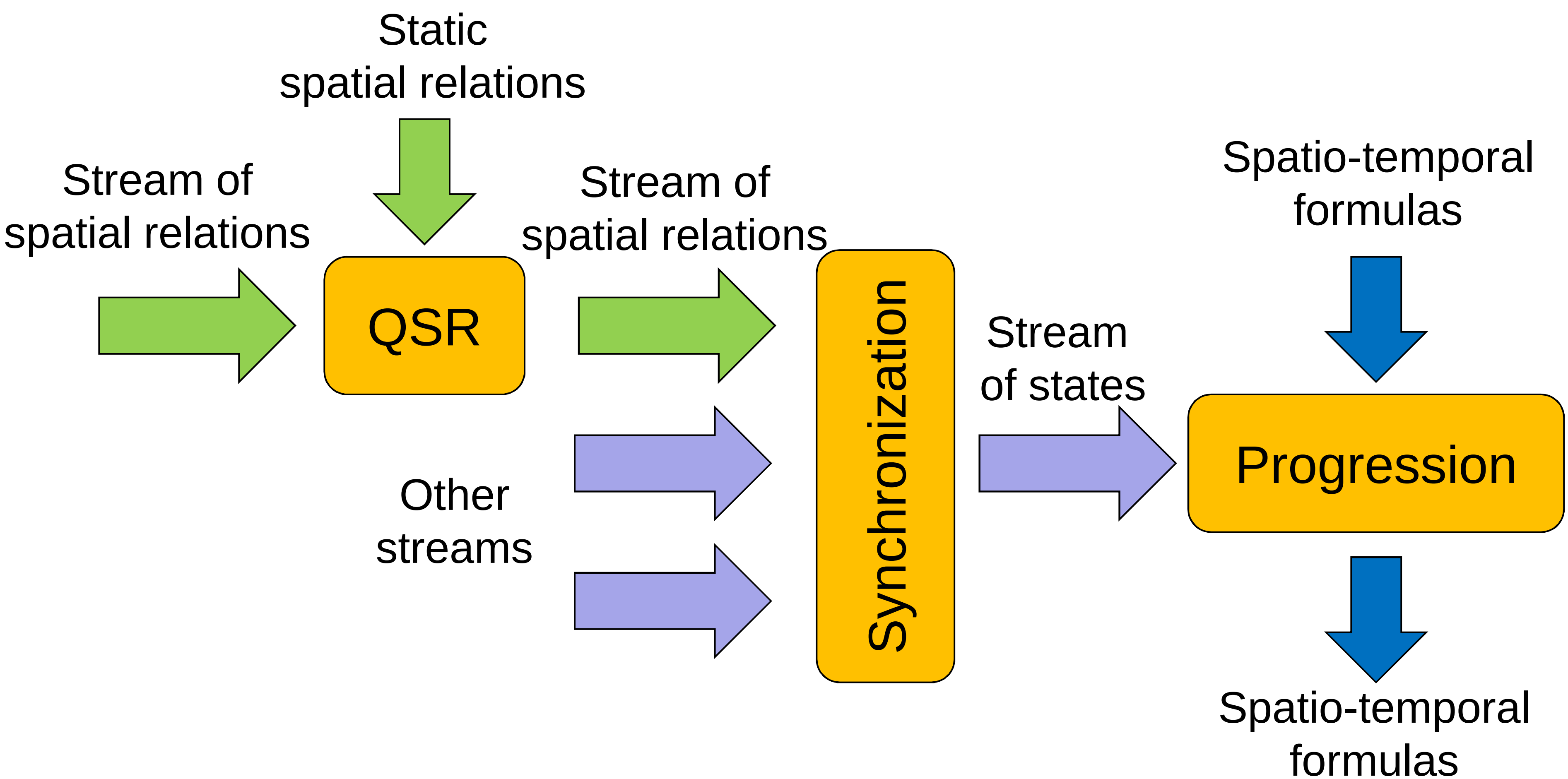
Known: EC(a,c) $\wedge$ NTPP(c,b)

Deduced: PO(a,b) $\vee$ TPP(a,b)
$\vee$ NTPP(a,b)

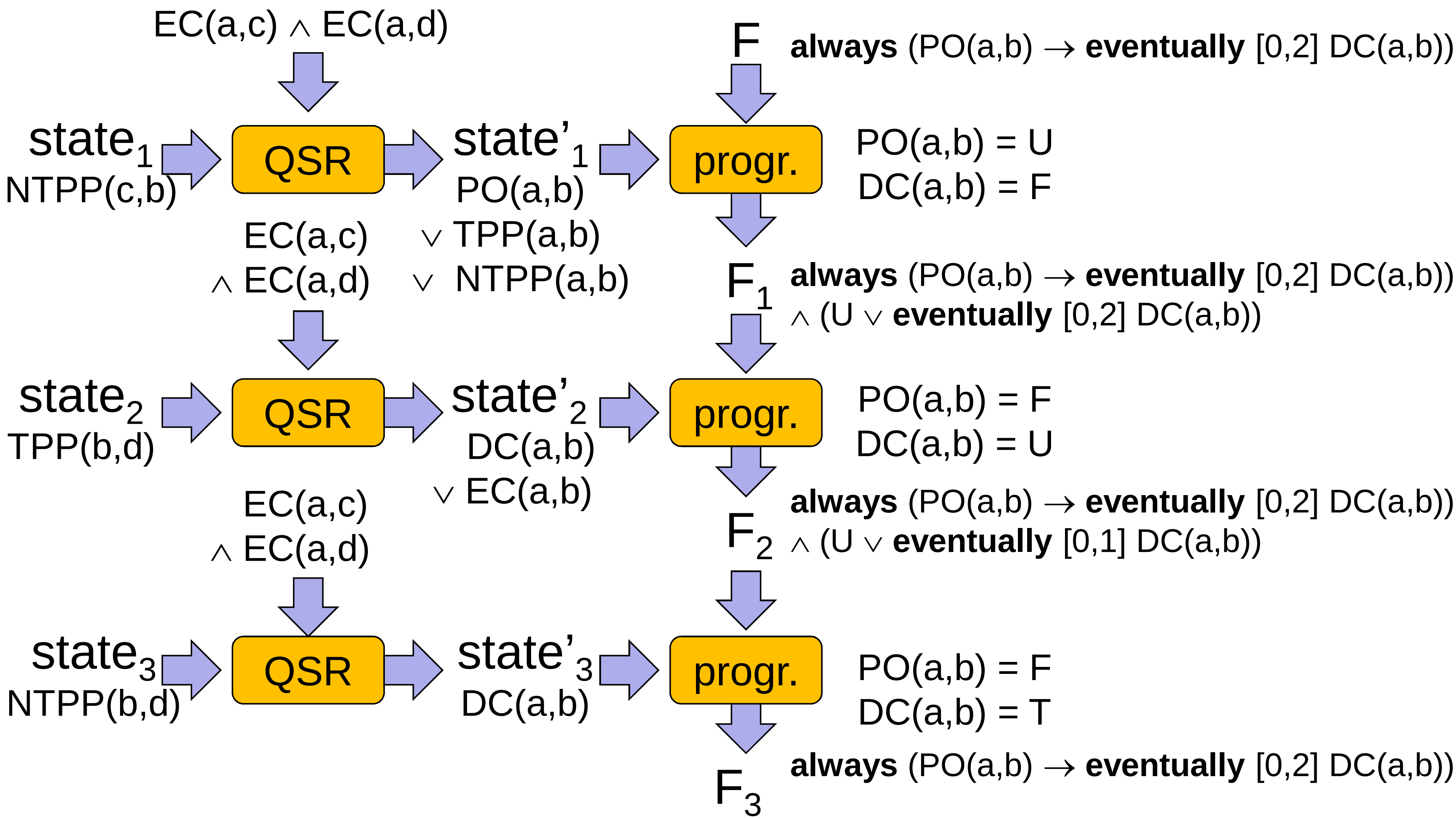
PO(a,b) = U and DC(a,b) = F

Spatio-Temporal Stream Reasoning in DyKnow

[Heintz and de Leng ECAI 2014]



Progression of Metric Spatio-Temporal Logic



The Sense-Reasoning Gap

Sense-Reasoning Gap

Reasoning

Qualitative models

Knowledge fusion

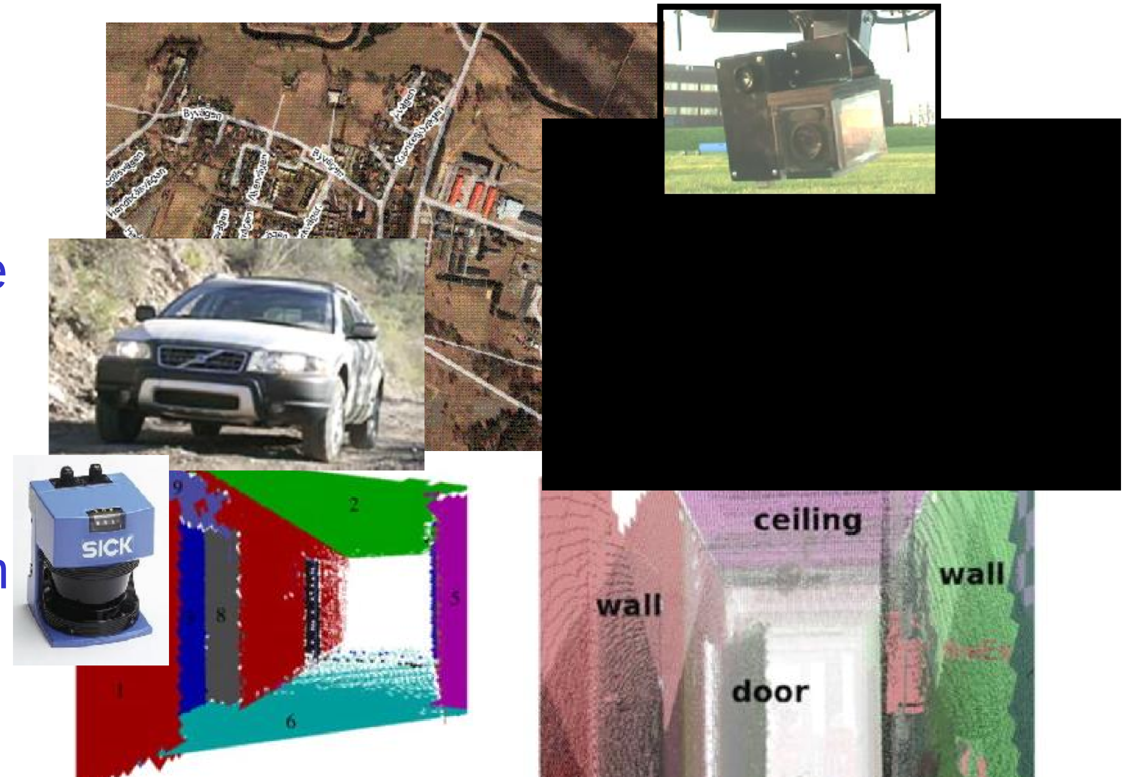
Mixed quantitative and qualitative models

Information fusion

Quantitative models

Sensor fusion

Sensing

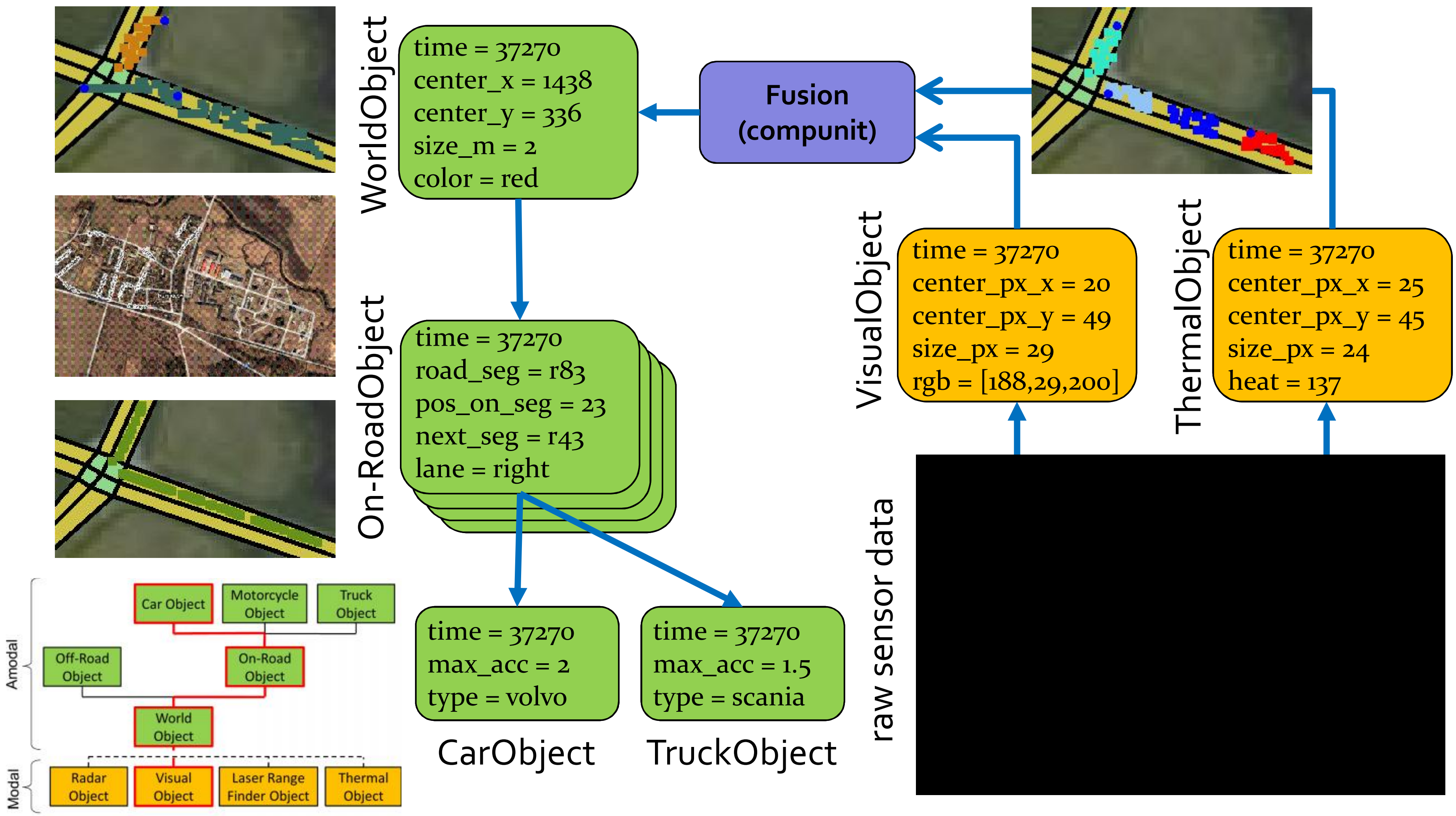


The ***sense-reasoning gap*** is the gap between the noisy numerical data directly generated by sensors and the crisp symbolic information that reasoning functionalities normally assume to be available.

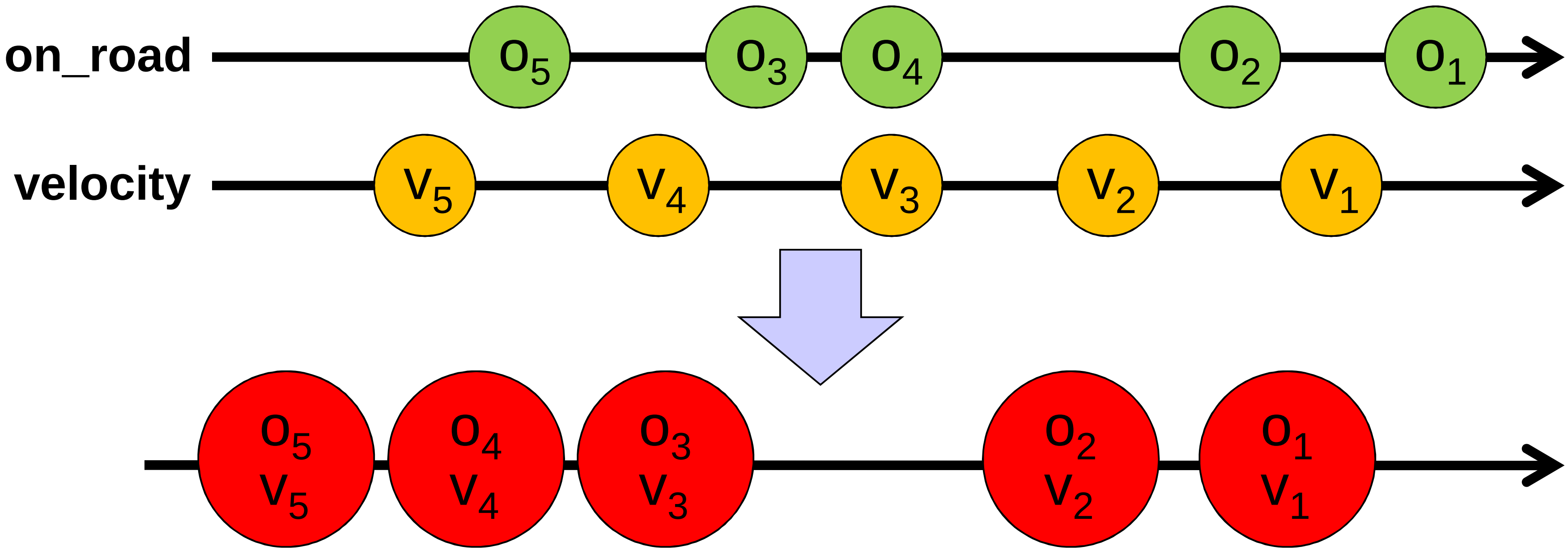
Linköpings universitet

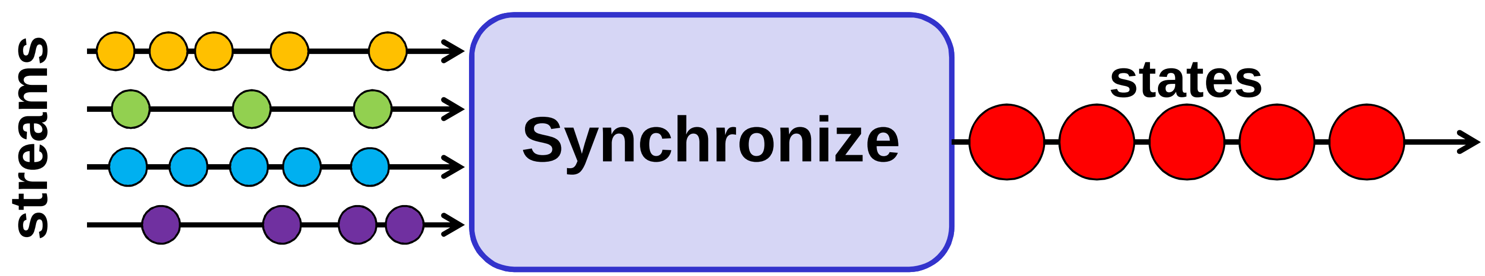
AIICS Autonomous UAV Technologies

Bridging the Sense-Reasoning Gap



Generating State Streams





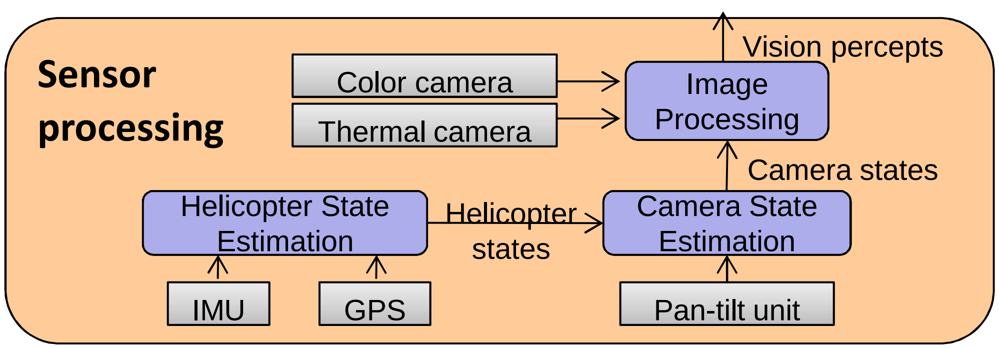
Grounding Stream Reasoning in Robotic Systems

[de Leng and Heintz FOfAI 2015, Heintz IROS 2014; de Leng and Heintz FUSION 2013]

- A temporal logical formula contains a number of symbols representing variables whose values over time must be collected and synchronized in order to determine the truth value of the formula.

  forall *x* in UAV **always**(Speed[*x*] < 60)

- Given a functional system, such as a robot, producing streams the grounding problem for logic-based stream reasoning is to *connect symbols in formulas to streams in the functional system so that the symbols get their intended meaning*.



Syntactic and Semantic Grounding

[de Leng and Heintz FOfAI 2015, Heintz IROS 2014; de Leng and Heintz FUSION 2013]

- *Syntactic grounding*: Use a direct mapping between symbols and streams, for example by using stream names in the formulas. A formula such as

  forall *x* in UAV **always**(Speed [x] < 60)

  would then have to be written something like

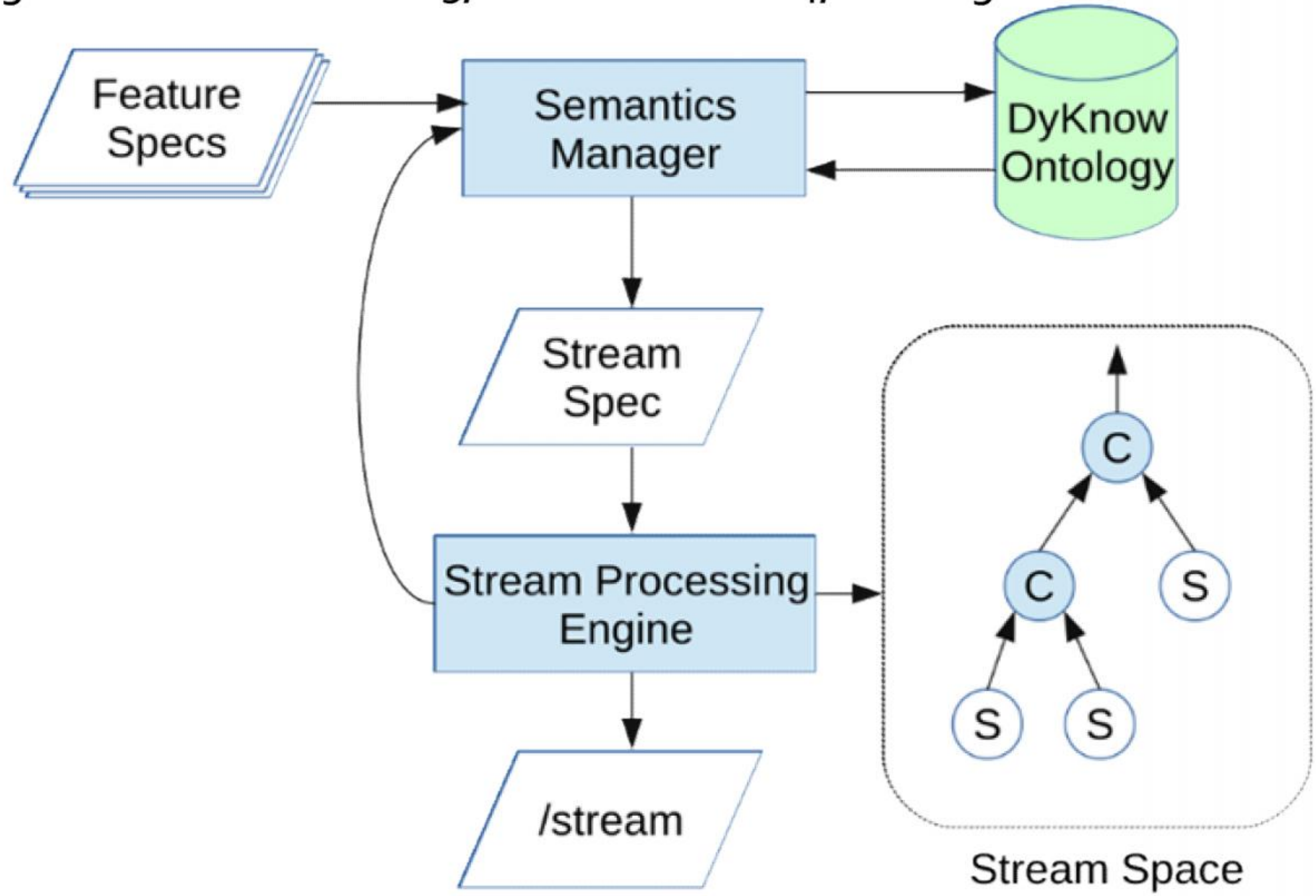
  always((/uav1/uavstate.spd < 60) ∧ (/uav2/uavstate.spd < 60))

- *Semantic grounding*: Annotate streams with their semantic content and reason about how to connect symbols to streams using semantic web technologies. We call this reasoning for *semantic matching*. It finds the relevant streams by matching the ontological concepts used in a formula to the ontological concepts associated with the streams.

Semantic Web

Semantically Grounded Stream Reasoning

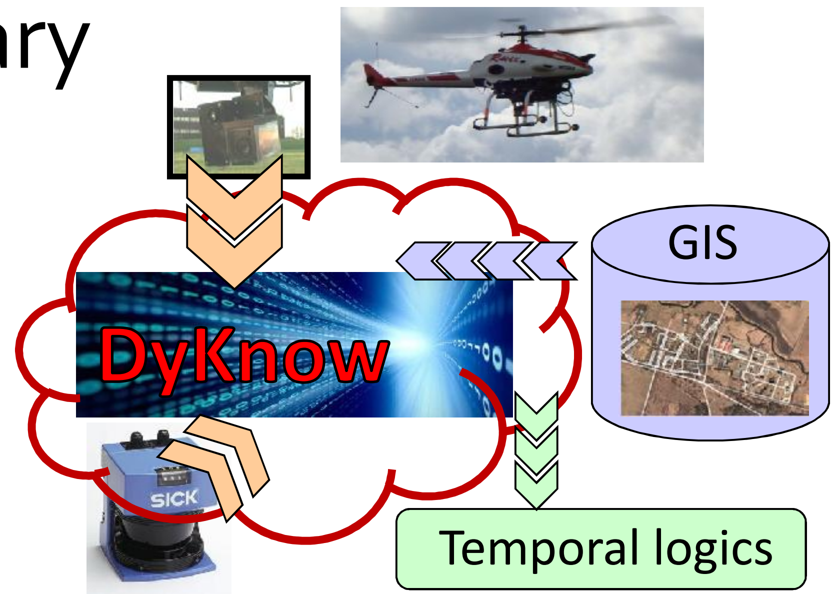
[de Leng and Heintz FOfAI 2015, Heintz IROS 2014; de Leng and Heintz FUSION 2013]



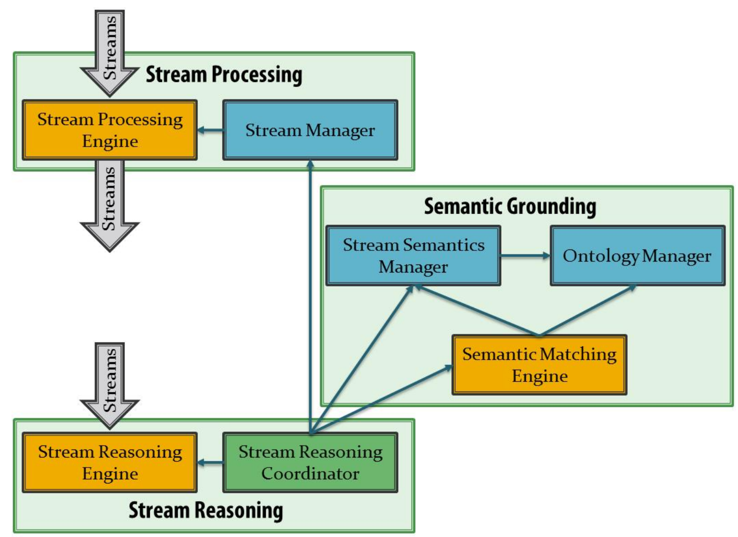
Discussion

- A stream can be seen as a partial view of a conceptual central KB. One important and interesting special case is to view a stream as temporal information, where each stream element is conceptually a separate time-point.
- Windows are an explicit construct to create independent parts
- Event processing vs Stream Reasoning.
- Separating stream processing and stream reasoning.
- Determining the truth value of a formula over a given stream or finding all substitutions which make the formula true.
- Finding all substreams which satisfies a formula.
- Combine formula evaluation (path checking) with theorem proving (model checking).

Summary



- High level incremental reasoning over streaming information is essential to autonomous systems.
- DyKnow is a practical framework for grounded stream reasoning including support for spatio-temporal reasoning.
- The reasoning is semantically grounded through a common ontology and a specification of the semantic content of streams relative to the ontology.
- Through DyKnow, ROS is extended with a powerful stream reasoning capability available to a wide range of robotic systems.



Linköpings universitet

expanding reality

www.liu.se

LiU EXPANDING REALITY