A Novel Architecture for Situation Awareness Systems

Franz Baader, Andreas Bauer, Peter Baumgartner, Anne Cregan, Alfredo Gabaldon, Krystian Ji, Kevin Lee, David Rajaratnam, Rolf Schwitter

TECHNISCHE UNIVERSITÄT DRESDEN

NICTA

UNSW THE UNIVERSITY OF NEW SOUTH WALES

UNIVERSIDADE NOVA DE LISBOA

ANU THE AUSTRALIAN NATIONAL UNIVERSITY

MACQUARIE UNIVERSITY SYDNEY ~ AUSTRALIA

Situation Awareness

Situation Awareness (SA) is concerned with the perception of elements of their meaning and the projection of their status in the near future
[Endsley, 1995]

- Support for decision making in complex, dynamic areas
- Aviation, Air Traffic Control, Power Plant Operations, Military Command and Control
- Mitigate risk of human error
- Own set of conferences, journals

Project Background

- Joint project with Australia's Defence Science and Technology Organisation (DSTO)
- DSTO approached NICTA for help to build a system for **higher-level** situation awareness based on automated reasoning techniques
  - Go beyond state-of-the-art
  - Run as a one year pilot project
- Project outcome: **SAIL**
  - "Situation Awareness by Inference and Logic"
  - Novel architecture and prototype implementation following a "knowledge-based" declarative approach

Atlantis Scenario

- Detailled information on an evolving conflict on Atlantis
  - Geographical and political
  - Operational (air corridors) and military (assets, capabilities)
  - Sensor data (radar), spy reports
- Challenge: to reconstruct/analyse the **event list**

+20	2000	75th Air Defence Squadron in Cambonga moves 8 x SA-10 and 8 x SA-12 to Eaglevista via rail and roads.
+21	2000	Task Group leaves North America home port (44N64W) in direction of Atlantis to a position 200 NM off Caltrop seaport (6330N 2730W) [1827 NM @ 15 kts = 122 hrs = 5 days 2 hours][33 hrs to reach Cape Race (495 NM)]
+22	1200	Blueland requests Task Group to escort the cargo from open sea to Celtic Straits.
+23	0500	Task Group waits for Cargo off Cape Race [4600N 5200W]
+23	1600	Cargo reaches Task Group off Cape Race.
+25	1200	Redland's A50-2 takes off from Becker-Bender AFB [5250N 2006W] and flies to Eaglevista.
+25	1320	2 x Su-24E (ECM) take off from Krupali and fly towards Deeland City and then to Eaglevista.

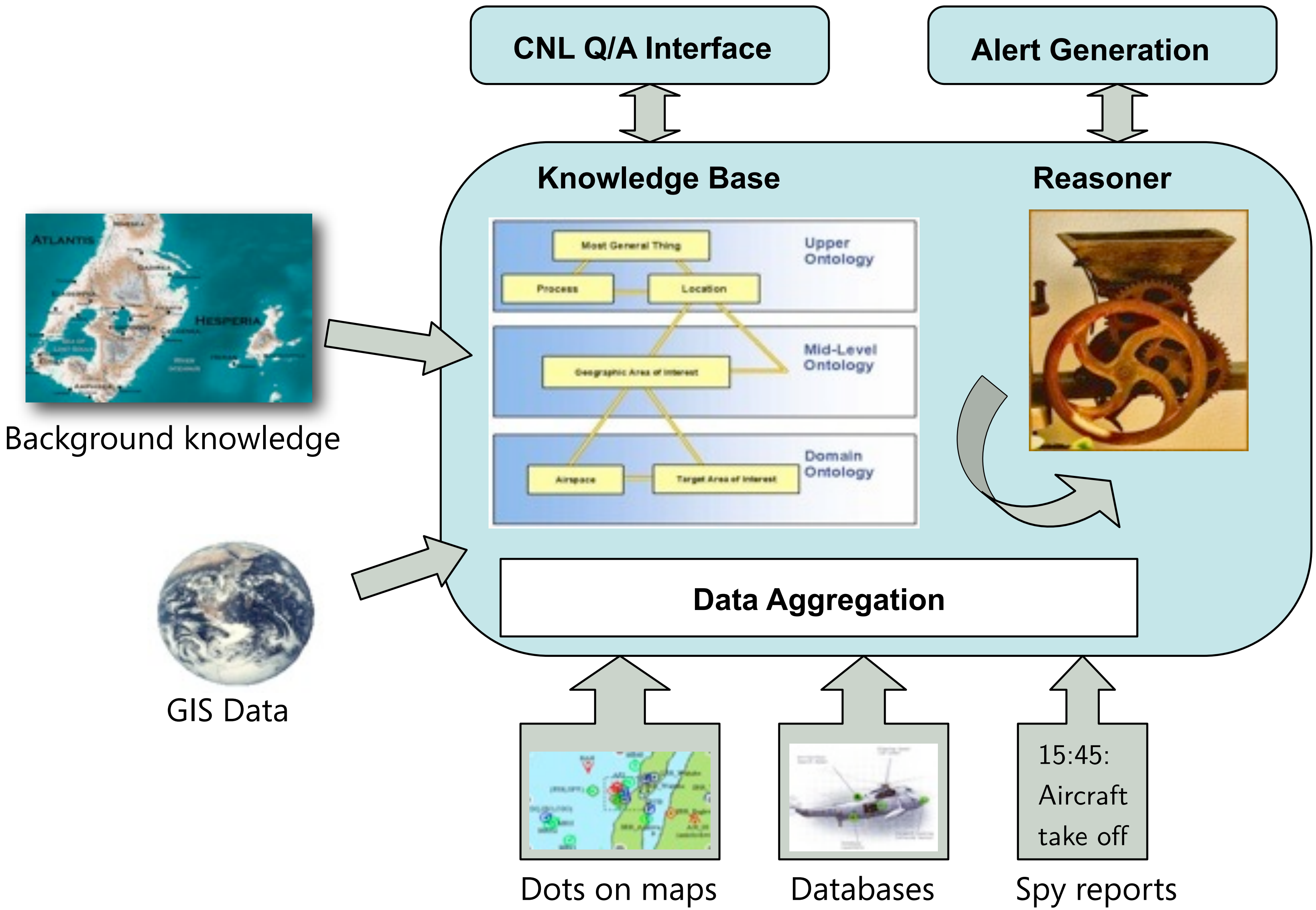
Higher-Level Situation Awareness

Q: What do these dots "mean"?

Higher-Level Situation Awareness

A: An Awacs surveilling a border, a greenpeace vessel

Combining Data/Information Sources



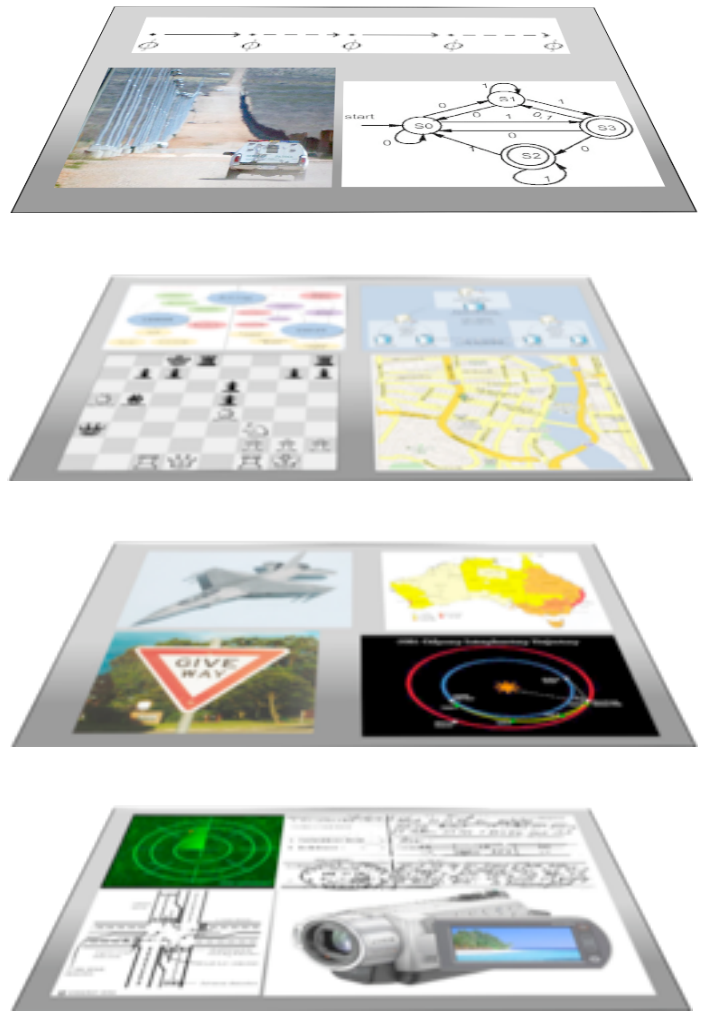
SAIL - System Architecture

Alert Generation
Identify Behaviour

Semantic Analysis
Identify Relations

Data Aggregation
Identify Entities

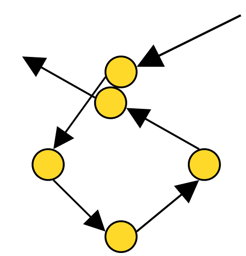
Collect Data



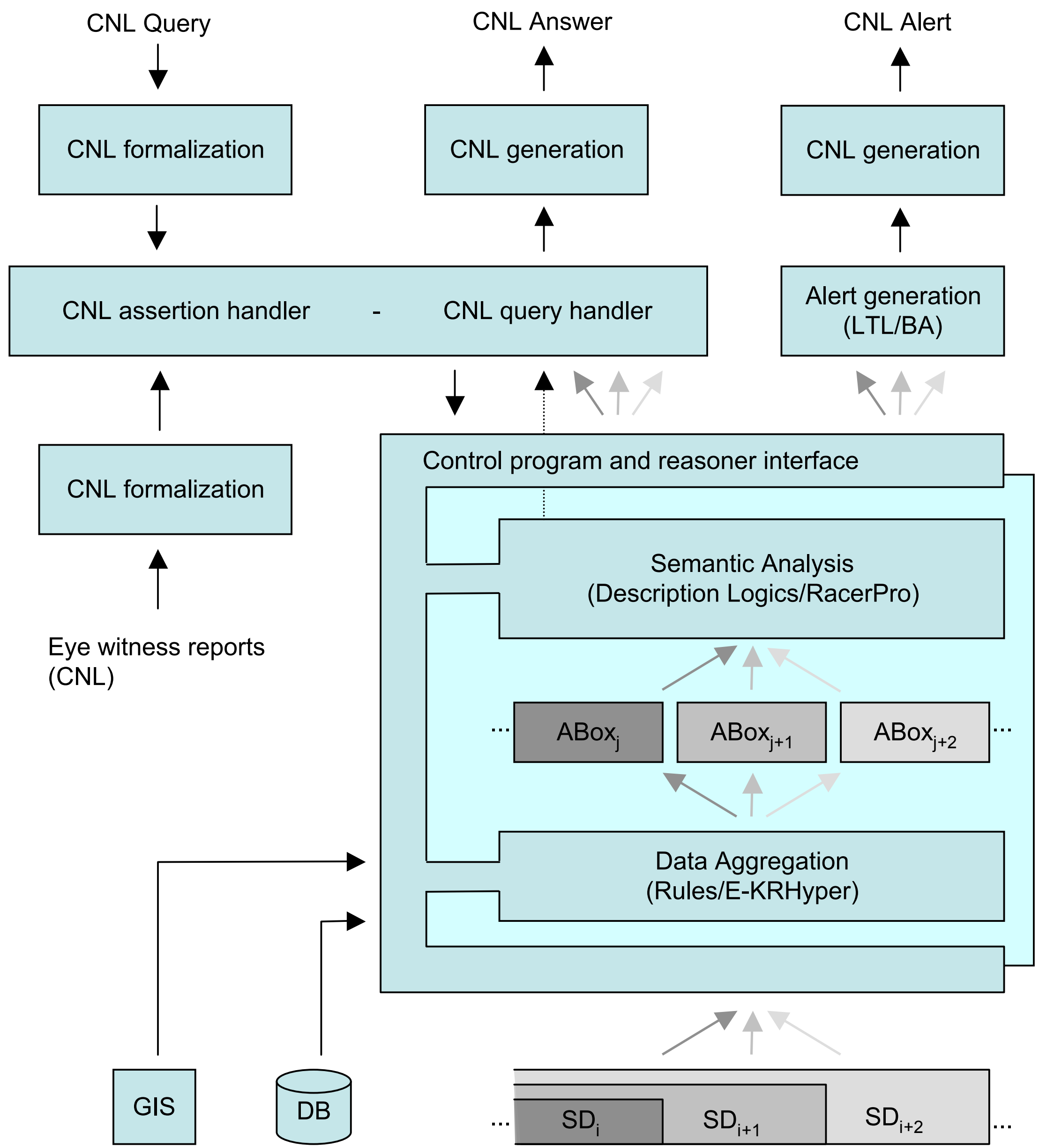
patrols(●, ●)

surveils(●, ●)

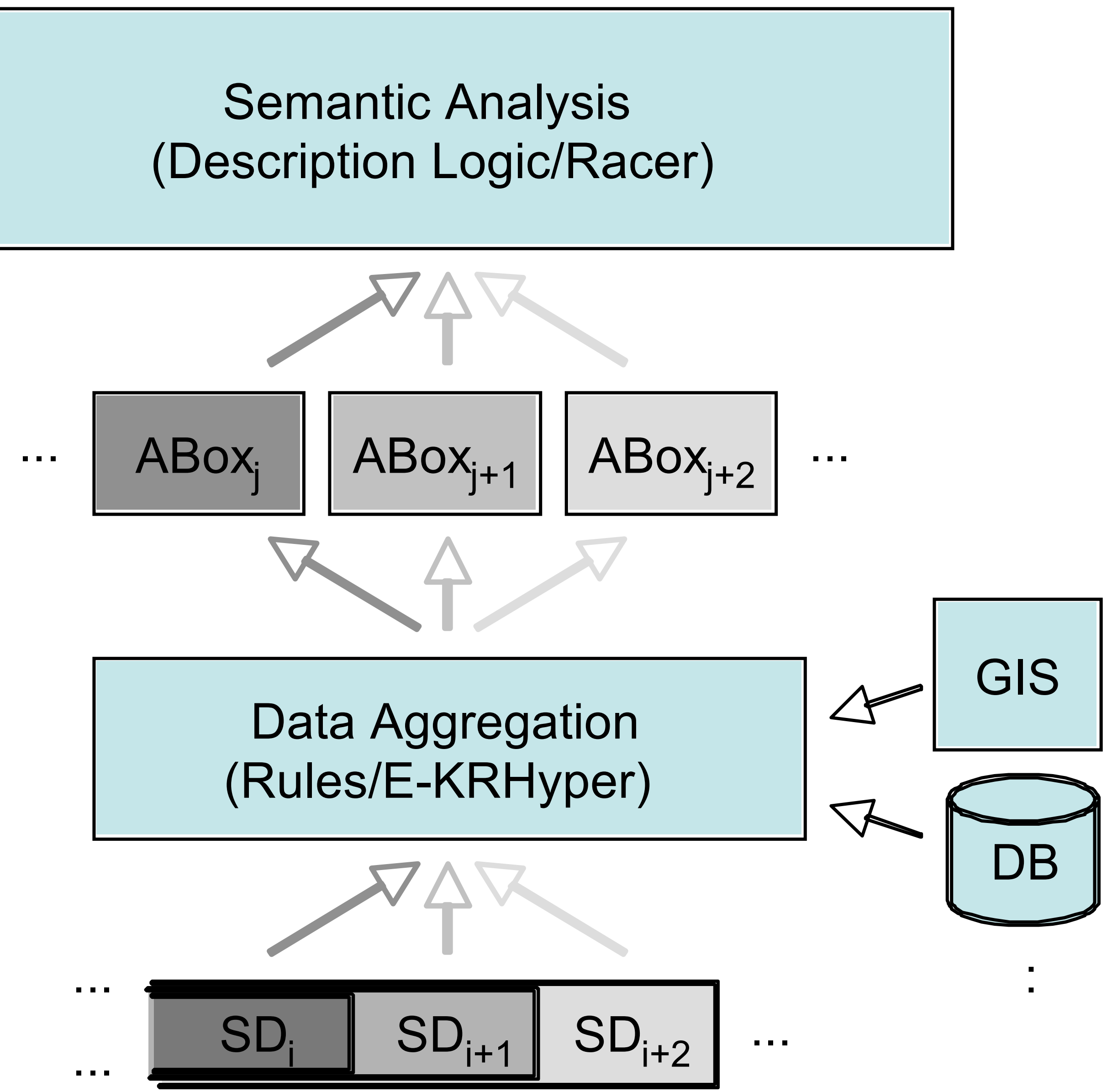
circle(●)



SAIL - System Architecture



Data Aggregation and Semantic Analysis



Data Aggregation

- Control program periodically invokes Data Aggregation layer on incoming Sensor Data (SD)
  - Maintains limited history of previous SD
- Data Aggregation layer analyse information over time
  - Detect capabilities: airstriker, surfacestriker
  - Synthesize events
- Specified as a disjunctive logic program (Rules)
  - Stratified default negation
  - Bottom-up evaluation, via KRHyper
  - Least model specifies an ABox

Data Aggregation Excerpt

```
object_appears(Obj, Now) :-
      current_time(Now),  % supplied by control program
      object(Obj, Now),   % Obj is in SD_Now
      previous_time(Now, T),
      \+ object(Obj, T).

take_off(Event, Obj, Now) :-
      object_appears(Obj, Now),
      in_air(Obj, Now),  % in_air computed by GIS
      concat(['ev_',Obj,'_',Now],Event).

%% assemble resulting ABox
abox(take_off(Event)) :- take_off(Event, Obj, Time).
abox(time(Even, Time)) :- take_off(Event, Obj, Time).
abox(object(Even, Obj)) :- take_off(Event, Obj, Time).
```

This is not Prolog
There is no "goal"

Some More Features

- Preserve information over time

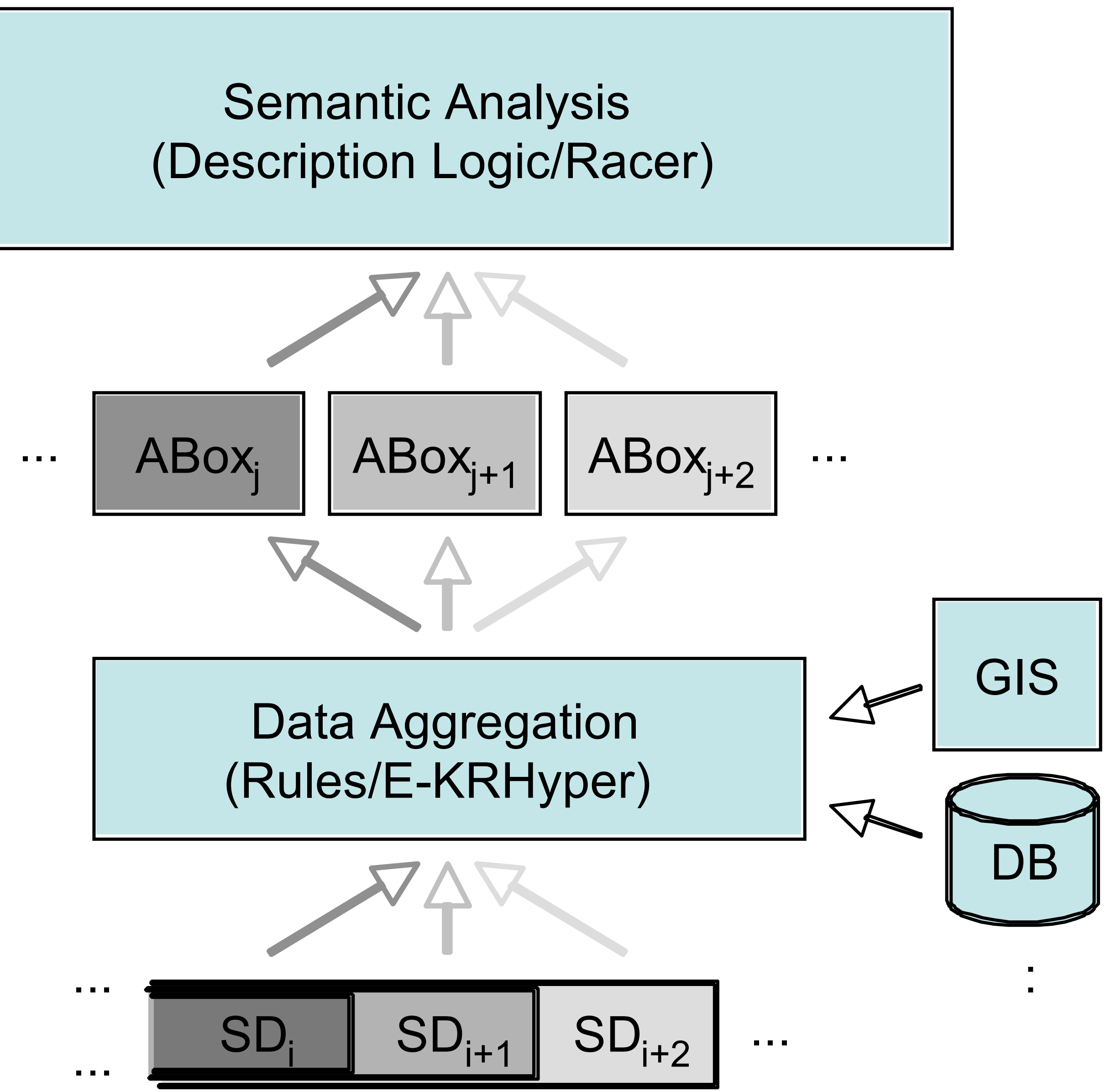
```
reassert(take_off(Event, Obj, CreationTime)) :-
   take_off(Event, Obj, CreationTime),
   current_time(Time),
   object(Obj, Time).  % reassert as long as Obj exists
```

- Can `Obj` reach `City`?

  For that, need to know

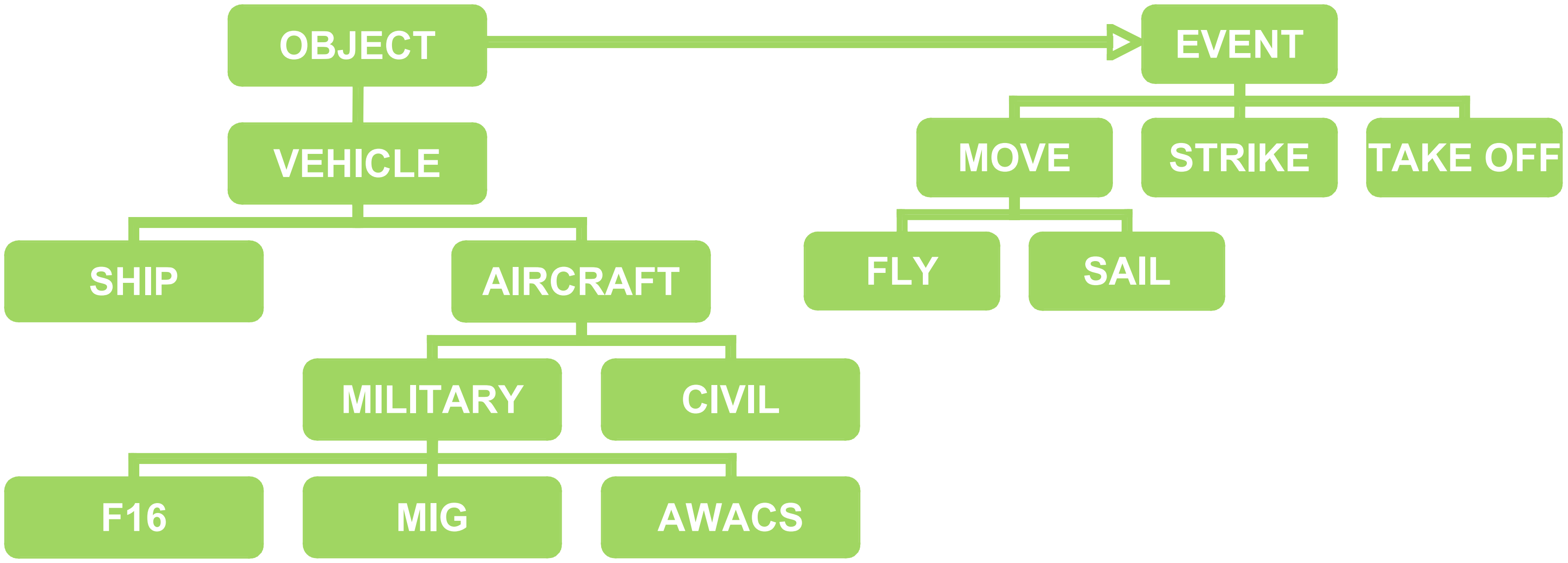
  - Distance between Obj and City, via GIS coupling
  - Time in air, kept in Data Aggregation layer
  - Aircraft capability, from database (facts)

Data Aggregation and Semantic Analysis



Semantic Analysis

- Conceptually higher level than Data Aggregation
  - Concepts e.g. aggresive, threat (holy grail)
  - Roles e.g. associated_with, enemy_organization
- Combines latest ABox and Description-Logic Ontology



Semantic Analysis

Ontology contains

$$\texttt{aggressive} \doteq \exists\ \texttt{has\_target}.\ (\texttt{physical\_object} \sqcup \texttt{space\_region})$$

Data Aggregation provides concept/role assertions

```
has_target(obj1, obj2).
physical_object(obj2).
```

It follows `aggressive(obj1)`

Could do this with Logic Program

Semantic Analysis

Ontology contains

$$\texttt{aggressive} \doteq \exists\ \texttt{has\_target}.\ (\texttt{physical\_object} \sqcup \texttt{space\_region})$$

An eye-witness report may provide (non-primitive) assertion

`aggressive(obj1).`

Can't do this (easily) with Logic Program

it follows that `obj1` has a target that is a `physical_object` or a `space_region`

Semantic Analysis

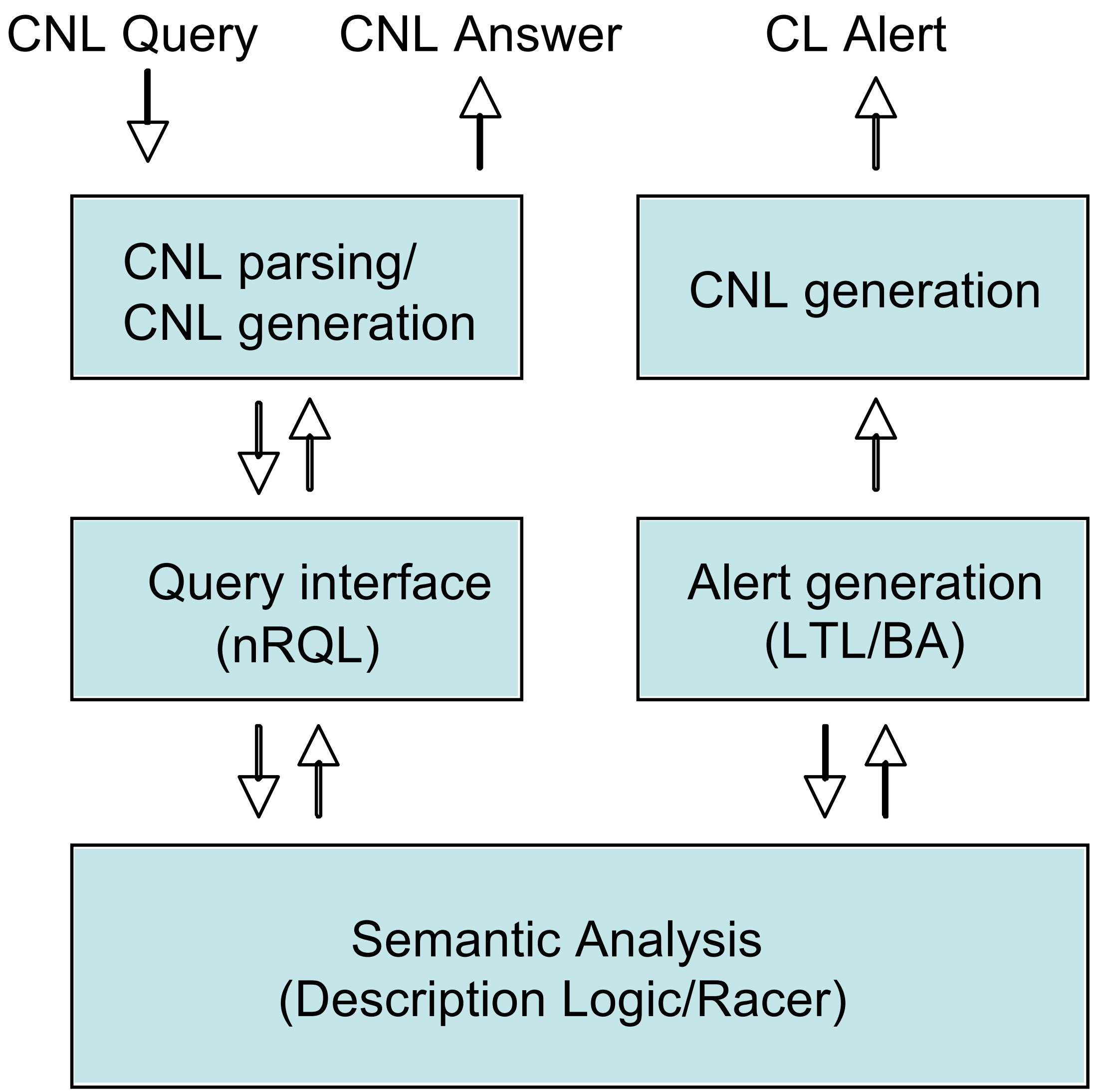
Non-primitive role assertions by means of nRQL rules

"a fighter associated with an enemy of blueland
targets an associate of blueland"

```
(firerule (and (?EM move)
               (?EM ?Ag has_theme)
               (?Ag fighter)
               (?Ag ?Org associated_with)
               (?Org s_blueland enemy_organization)
               (?EM ?Y has_direction)
               (?Y s_blueland associated_with))
   ((related (new-ind aggr ?Ag ?Y) ?Y has_target)))
```

Creates a new individual `aggr-Ag-Y` that is in the `has_target` relation with `Y`

Alert Generation



Alert Generation

- Capture a critical situation
- Are raised automatically by system
- Formally defined via linear temporal logic (LTL)

$$\mathbf{G}(\neg \mathit{aggressive}(p))$$

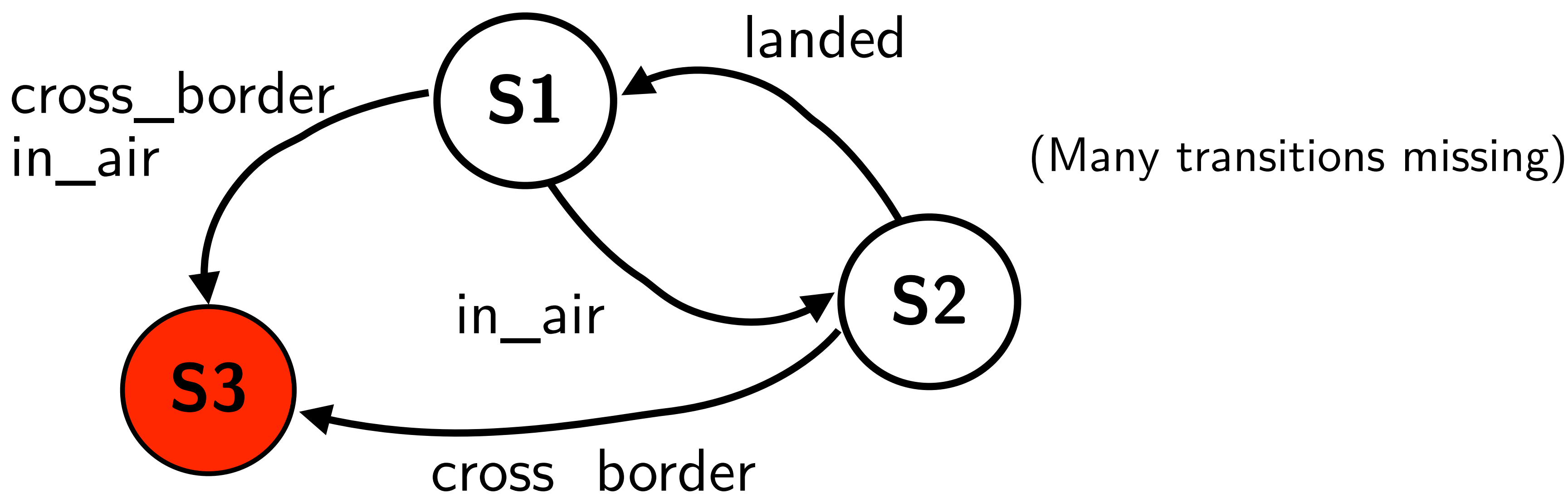
``If we detect that an enemy aircraft has taken off, and if this aircraft crosses our border, an alarm signal should be raised.''

$$\varphi := \mathbf{G}(\mathit{in\_air}(p) \Rightarrow \neg \mathit{cross\_border}(p)\ \mathbf{U}\ \mathit{landed}(p)).$$

- Caveat: no reasoners for temporal descripton logics available

Alert Generation by Runtime Verification

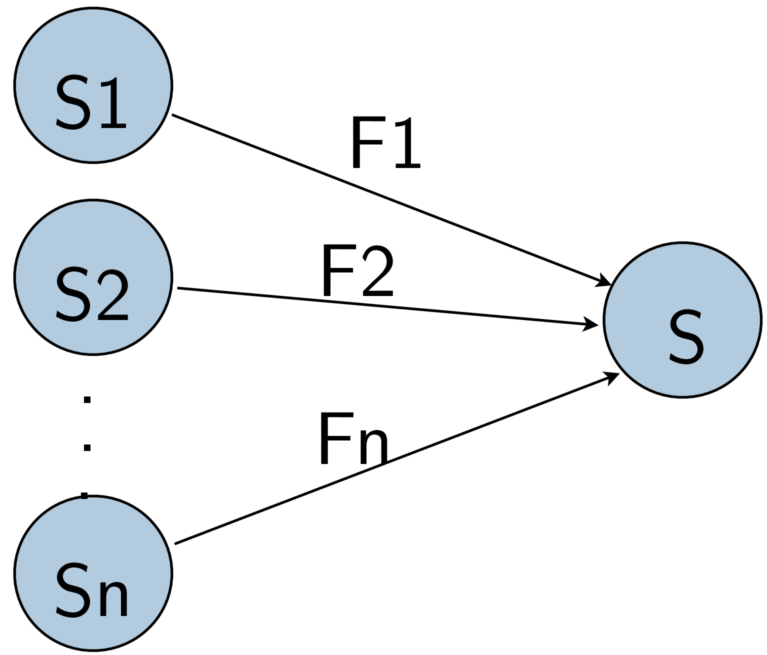
- Operationalization of a LTL formula ϕ in terms of a monitor
- Monitor is a Finite State Machine that reads a finite prefix $u \in \Sigma^*$ and determines if ($w \in \Sigma^\omega$)
  - for all w: $uw \models \phi$ (u is a good prefix), or
  - for all w: $uw \not\models \phi$ (u is a bad prefix)
- Otherwise there exists w, w' such that $uw \models \phi$ and $uw' \not\models \phi$



$$\varphi := \mathbf{G}(\mathit{in\_air}(p) \Rightarrow \neg \mathit{cross\_border}(p)\ \mathbf{U}\ \mathit{landed}(p)).$$

Alerts via Description Logic

FSM monitor: state S
and predecessors



Time: 1 2 . . . j-1 j . . .

$$\bigsqcup_i C_{S_i} \sqcap F_i \Rightarrow C_S$$

For each state S

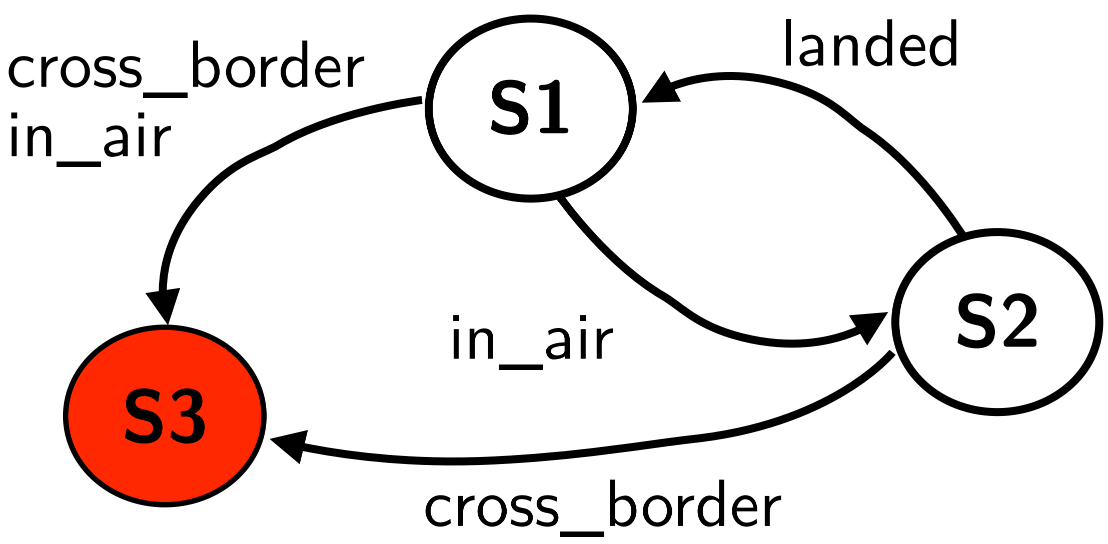
For individuals that satisfy LHS query,
populate concept Cs in the next time point.

Alerts: integration via DL reasoner

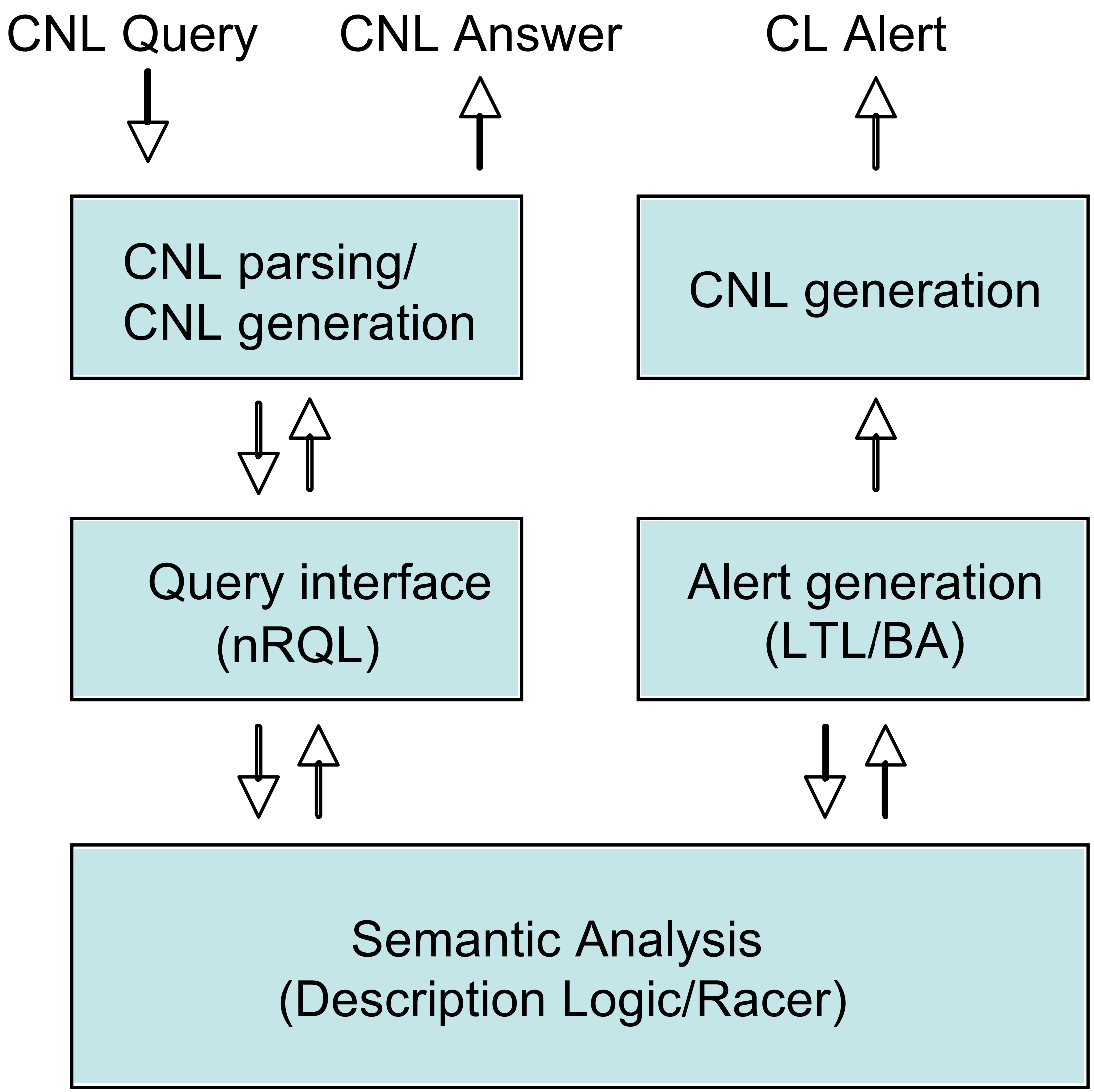
Example: $\varphi := \mathbf{G}(\mathit{in\_air}(p) \Rightarrow \neg \mathit{cross\_border}(p)\ \mathbf{U}\ \mathit{landed}(p)).$

Corresponding concepts/queries for three states:

$$
\begin{aligned}
C_1 : &\quad (C_1 \sqcap \neg \mathit{in\_air}) \sqcup (C_2 \sqcap \mathit{landed}),\\
C_2 : &\quad (C_2 \sqcap \neg \mathit{landed} \sqcap \neg \mathit{cross\_border}) \sqcup (C_1 \sqcap \mathit{in\_air}),\\
C_3 : &\quad C_3 \sqcup (C_1 \sqcap \mathit{cross\_border}) \sqcup (C_2 \sqcap \mathit{cross\_border}).
\end{aligned}
$$



Controlled Natural Language



Controlled Natural Language

- A CNL is an engineered subset of a natural language
  - It looks like English but it is a formal language
- CNL serves as a high-level interface language to SAIL
- Usage:
  - Add eye-witness reports (akin to sensor data)
  - Query the DL knowledge base
- Design of DL knowledge has to be "compatible" with CNL query language
  - events, thematic roles

Controlled Natural Language

- Eye-witness reports

  SU_24M takes off from Becker-Bender at 09:00.
  The A50-1 takes off from Krupali at 09:30.
  The fighter (SU_24M) flies towards Bendeguz.
  The AWACS (A50-1) flies towards Eaglevista.

  are translated into TPTP and then added to the KB
- Anaphora are resolved with the help of the DL ontology

Controlled Natural Language

- Queries

  What aircraft of Redland is able to reach a city of Blueland?

  are translated into conjunctive nRQL queries:

```
(retrieve (?1)
   (and (?1 aircraft)
        (?1 s_redland associated_with)
        (?2 ?1 has_agent)
        (?2 reach)
        (?2 ?3 has_theme)
        (?3 city)
        (?3 s_blueland associated_with)))
```

  and answers are generated in CNL

Conclusions

- SAIL: Layered architecture based on different logical formalisms
  - Tableaux-based answer-set programming (data aggregation)
  - Description logic (semantic analysis)
  - Temporal logic (alert generation)
- System is implemented
  - Tested with excerpts from "Atlantis Scenario"
  - Google Earth interface, GIS system
- Short project runtime of 1 year
  - Work with existing automated reasoning systems
  - Successor project did not happen