Can Logical Reasoning be Used to Achieve Higher-Level Situation Awareness?

Franz Baader, TU Dresden, Germany

Levels of Situation Awareness

[Endsley; 1995]

Projection — Projecting how events may evolve over time, guess future behaviour

↑

Comprehension — Higher-level composite view of situation, at semantic level of human comprehension

forrest

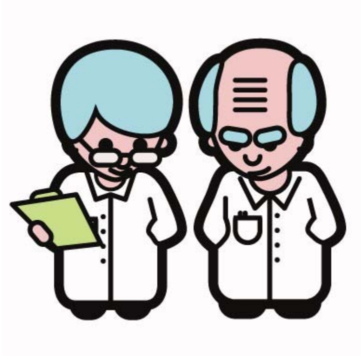
↑

Perception — Sensors, Radar, Pictures, Videos, Eyewitness reports by humans

lots of trees

© Franz Baader

Levels of Situation Awareness



Projection

danger of myocardial infarction

medical ontology | patient record

Comprehension

hypertension | history of hypertension

Perception

diastolic arterial pressure = 188,79

red face

Dresden

© Franz Baader

Levels of Situation Awareness

the role of logic

Projection

↑

Comprehension

↑

Perception

Higher-level composite view expressed in logic

Background facts

Background theory

- declarative semantics independent of implementation
- reasoning can be used to derive consequences
- even if the represented knowledge is incomplete

Dresden

© Franz Baader

Problems to be solved in this context

- Which logic should we use for the representation?
  - Expressiveness versus efficiency of reasoning.
  - Reasoning about data that change over time.
- How can the raw data from the preception layer be transformed into a logical representation?
  - without losing the advantage of a declarative approach.
- How can users that are not trained in formal logic interact with the system?
  - add high-level information to the knowledge base built in the comprehension layer; query this knowledge base.
- How can critical situations be monitored without requiring user interaction?

© Franz Baader

What have we done until now to solve these problems?

- Opening doors using Description Logic ontologies.
  - Six-months pilot project funded by Siemens in the context of the intelligent house.
  - Built small ontology modelling different contexts for an intelligent door.
  - Tested whether current Description Logic reasoners are efficient enough to answer context queries.
- Use medical ontologies expressed in Description Logics to monitor the medical status of patients.
  - part of a two-year basic research project on integrating Description Logics and action languages (funded by DFG)
  - developed new temporalized Description Logic for which reasoning is more efficient than in previously proposed such logics
  - showed that this logic can be used to generate monitors (finite state automata with output)
- The SAIL project.

Dresden

© Franz Baader

SAIL

Situation Awareness by Inference and Logic

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

Project Background

- Joint project between National ICT Australia (NICTA) and 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
- Outcome of SAIL project:
  - Novel architecture and prototype implementation following a knowledge-based declarative approach
  - Prototypical implementation of system that employs existing reasoners and public-domain GIS system

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.

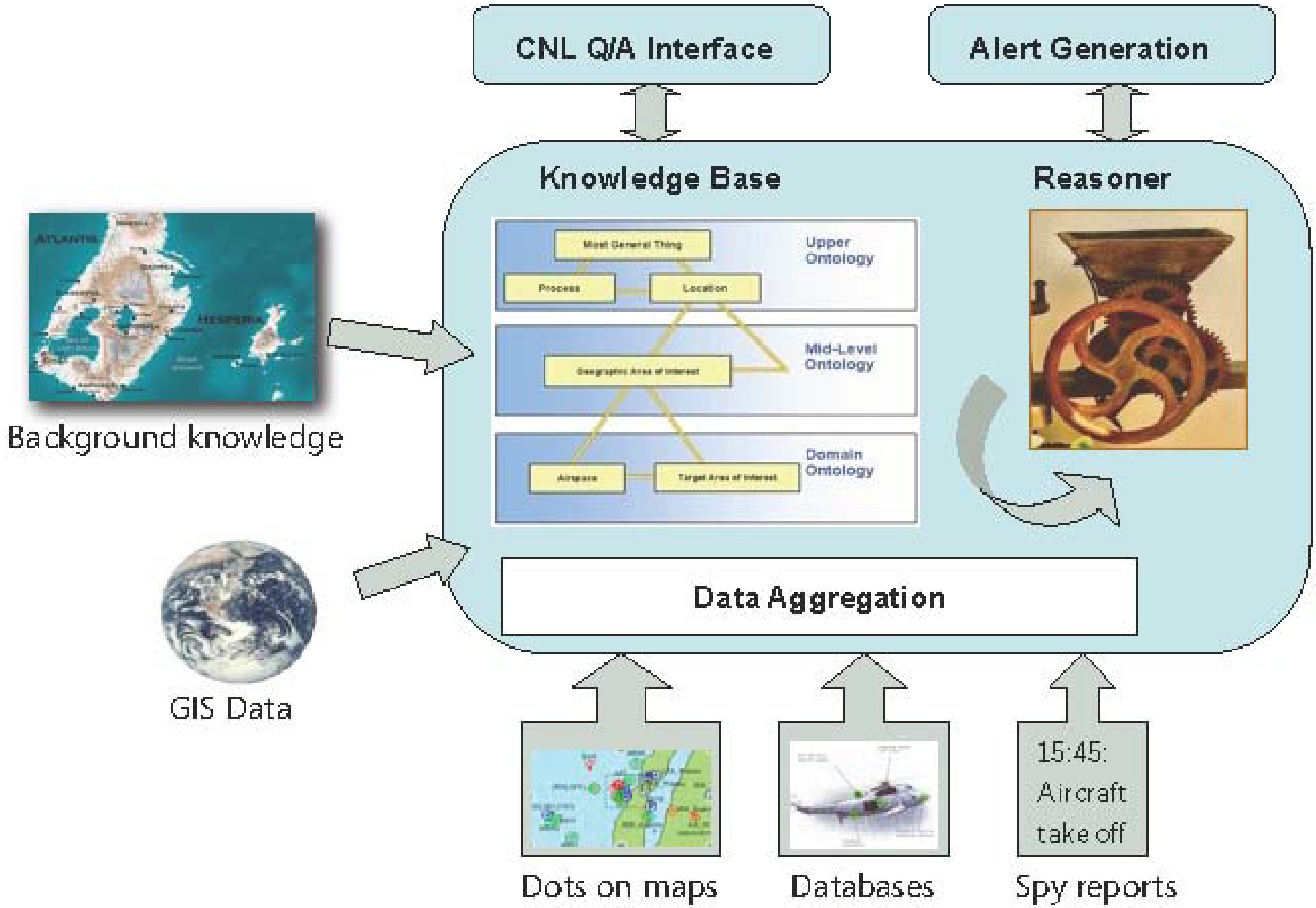
Higher-Level Situation Awareness

Q: What do these dots "mean"?

Higher-Level Situation Awareness

A: An Awacs surveilling a border, a greenpeace vessel

Combining Data/Information Sources



Problems

How did we address them in SAIL?

- Which logic should we use for the representation?
  - Expressiveness versus efficiency of reasoning.

Description Logics offer a good compromise.

© Franz Baader

research of the last 20 years

Phase 1:

- implementation of systems (Back, K-Rep, Loom, Meson, . . .)
- based on incomplete structural subsumption algorithms

Phase 2:

- development of tableau-based algorithms and complexity results
- first implementation of tableau-based systems (Kris, Crack)
- first formal investigation of optimization methods

Phase 3:

- tableau-based algorithms for very expressive DLs
- highly optimized tableau-based systems (FaCT, Racer)
- relationship to modal logic and decidable fragments of FOL

Phase 4:

- Web Ontology Language (OWL-DL) based on very expressive DL
- industrial-strength reasoners and ontology editors for OWL-DL
- investigation of leight-weight DLs with tractable reasoning problems

© Franz Baader

Description logic system

structure

description language

- constructors for building complex concepts out of atomic concepts and roles
- formal, logic-based semantics

knowledge base

TBox

defines the terminology of the application domain

ABox

states facts about a specific “world”

reasoning component

- derive implicitly respresented knowledge (e.g., subsumption)
- “practical” algorithms

© Franz Baader

Description language

Constructors of the DL $\mathcal{ALCN}$:

$C \sqcap D, C \sqcup D, \neg C, \forall r.C, \exists r.C, (\geq n\, r), (\leq n\, r)$

A man	$Human \sqcap \neg Female \sqcap$
that has a rich or beautiful wife	$\exists married\_to.(Rich \sqcup Beautiful) \sqcap$
and at least 3 children,	$(\geq 3\ child) \sqcap$
all of whom are happy	$\forall child.Happy$

TBox

definition of concepts

$Happy\_man \equiv Human \sqcap \ldots$

more complex constraints

$\exists child.Human \sqsubseteq Human$

$Professor \sqsubseteq \neg Rich$

ABox

properties of individuals

$Happy\_man(Franz)$

$married\_to(Franz, Inge)$

$child(Franz, Luisa)$

Dresden

© Franz Baader

Problems

How did we address them in SAIL?

- Which logic should we use for the representation?
  - Expressiveness versus efficiency of reasoning.

Description Logics offer a good compromise. ?

  - Reasoning about data that change over time.

Reasoning in temporalized DLs is of a very high complexity.

  - partially deal with temporal information on the data aggregation layer
  - produce time-stamped ABoxes
  - use time-stamps in queries (but not in ontology)

© Franz Baader

Problems

How did we address them in SAIL?

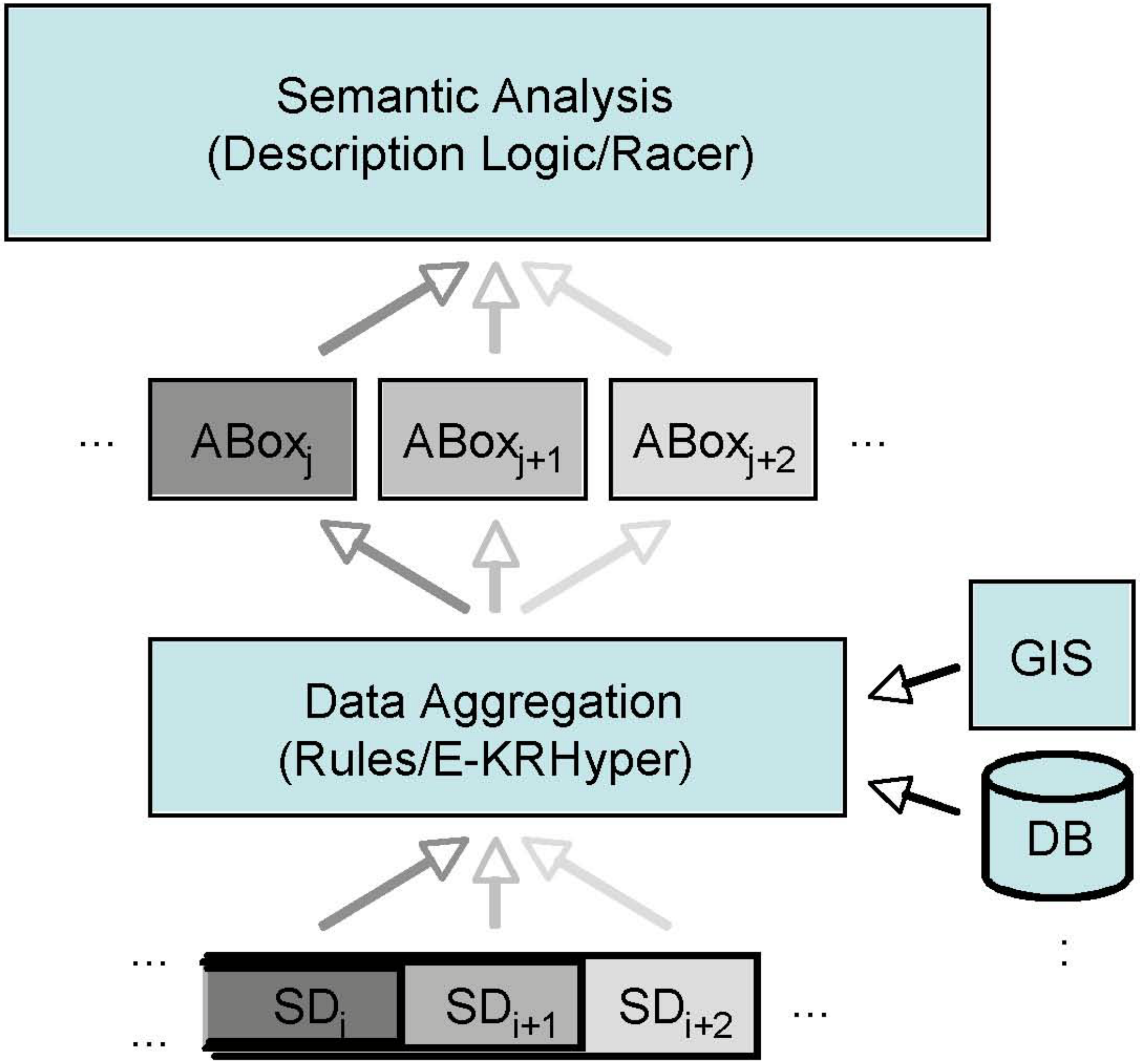
- How can the raw data from the preception layer be transformed into a logical representation?
  - without losing the advantage of a declarative approach.

Connection between the data aggregation layer and the semantic layer is achieved through primitive concepts/roles

- are not fully defined in the ontology
- populated by instances computed in the data aggregation layer
- rules/programs filling the primitives need to be sound w.r.t. their intended meaning in the application domain, but not complete
- DL reasoner can then also deduce assertions involving defined concepts/roles
- eye-witness reports may directly yield assertions for defined concepts/roles

© Franz Baader

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

Use of logic-based programming language not vital,
but declarative approach allows for easier understanding

Data Aggregation Excerpt

```
object_appears(Obj, Now) :-
      current_time(Now),  % supplied by control program
      object(Obj, Now),   % Obj is in SDNow
      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"

Semantic Analysis

Ontology contains

```
aggressive ≐ ∃ has_target.
     (physical_object ⊔ space_region)
```

Data Aggregation provides concept/role assertions

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

It follows `aggressive(obj1)`

Problems

How did we address them in SAIL?

- How can users that are not trained in formal logic interact with the system?
  - add high-level information to the knowledge base built in the comprehension layer; query this knowledge base.

Use Controlled Natural Language (CNL)

- engineered subset of a natural language that looks like English, but has restricted syntax and formal semantics
- used as high-level interface language for the SAIL system: add eye-witness reports; query the DL knowledge base
- designed such that questions can be translated into nRQL queries over the SAIL ontology

supported by DL system RACER

© Franz Baader

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

Problems

How did we address them in SAIL?

- How can critical situations be monitored without requiring user interaction?

Alert generation:

- alerts capture a critical situation;
- are automatically generated by the system;
- formally specified in DL extension of linear temporal logic (LTL) [B., Lutz, Ghilardi; 2008];
- extends approach for generating monitors in runtime verification for propositional LTL [Bauer, Leucker, Schallhart; 2006]

Current system:

ad hoc integration with DL reasoner

Formal approach described in [B., Bauer, Lippmann; 2009]

Dresden

© Franz Baader

Problems

How did we address them in SAIL?

- How can critical situations be monitored without requiring user interaction?

Alert generation:

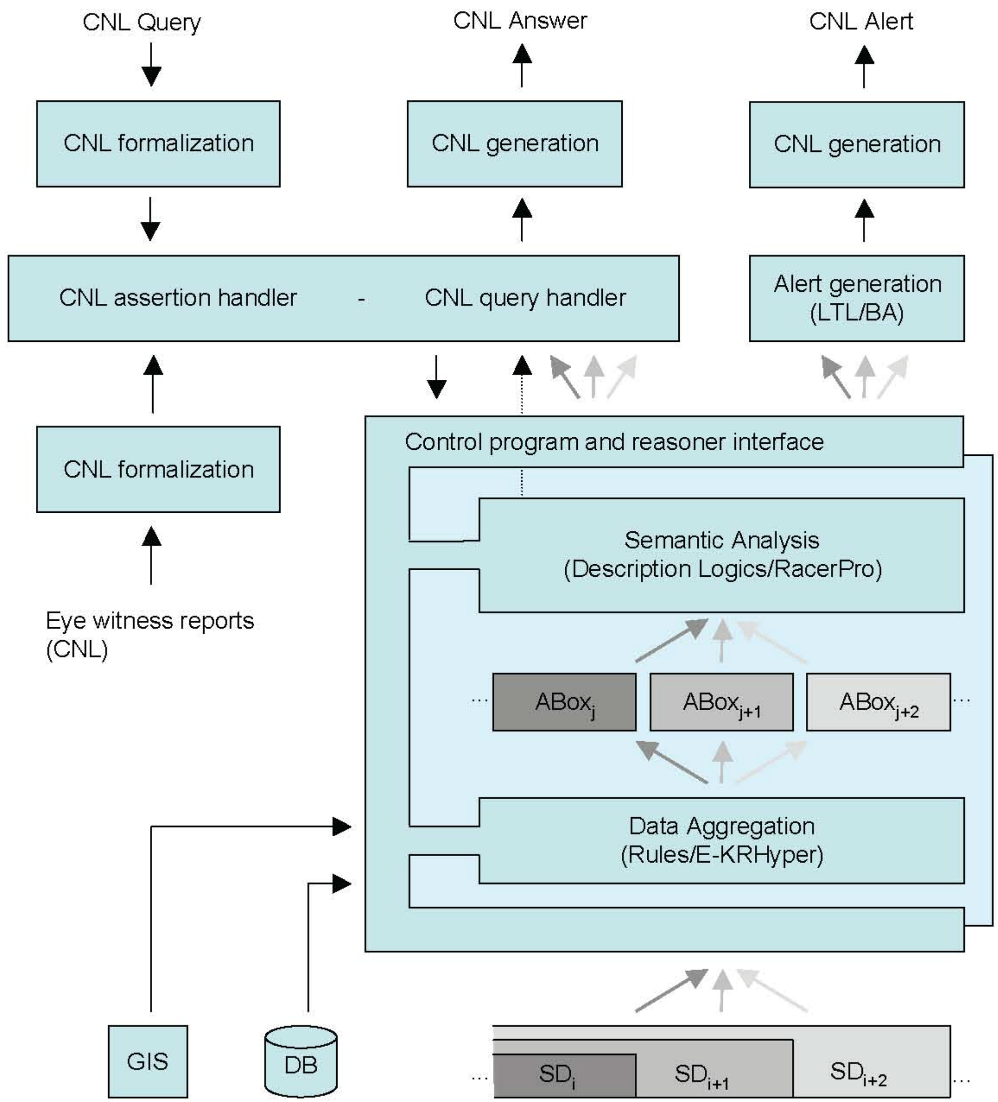
$$\mathbf{G}(\neg aggressive(p))$$

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

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

© Franz Baader

SAIL - System Architecture



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

Triggered new theory:

new DL-extension of LTL

runtime verification for this logic

not yet implemented


Questions?