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
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

- Detailed 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

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+20</td>
<td>2000</td>
<td>75th Air Defence Squadron in Cambonga moves 8 x SA-10 and 8 x SA-12 to Eaglevista via rail and roads.</td>
</tr>
<tr>
<td>+21</td>
<td>2000</td>
<td>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)]</td>
</tr>
<tr>
<td>+22</td>
<td>1200</td>
<td>Blueland requests Task Group to escort the cargo from open sea to Celtic Straits.</td>
</tr>
<tr>
<td>+23</td>
<td>0500</td>
<td>Task Group waits for Cargo off Cape Race [4600N 5200W]</td>
</tr>
<tr>
<td>+23</td>
<td>1600</td>
<td>Cargo reaches Task Group off Cape Race.</td>
</tr>
<tr>
<td>+25</td>
<td>1200</td>
<td>Redland’s A50-2 takes off from Becker-Bender AFB [5250N 2006W] and flies to Eaglevista.</td>
</tr>
<tr>
<td>+25</td>
<td>1320</td>
<td>2 x Su-24E (ECM) take off from Krupali and fly towards Deeland City and then to Eaglevista.</td>
</tr>
</tbody>
</table>
Q: What do these dots "mean"?
Higher-Level Situation Awareness

A: An Awacs surveilling a border, a greenpeace vessel
Combining Data/Information Sources

- **Knowledge Base**
  - **Reasoner**
  - **Data Aggregation**
  - **Alert Generation**
  - **CNL Q/A Interface**

Background knowledge

GIS Data

- Dots on maps
- Databases
- Spy reports

15:45: Aircraft take off
SAIL - System Architecture

Alert Generation
Identify Behaviour

Semantic Analysis
Identify Relations

Data Aggregation
Identify Entities

Collect Data

patrols(●, ○)
surveils(●, ○)
circle(●)
SAIL - System Architecture

Fig 0 – SAIL system architecture

Abbreviations:
- GIS: GIS system
- SD: sensor data
- LTL: linear temporal logic
- BA: B¨uchi automaton
- CNL: controlled natural language

Concerns about objects: SAIL accumulates this information: at each time point \( i \) the system stores data from previous time points \( SD_j \) for \( j \leq i \). For practical reasons — to cope with the amount of data accumulating over time — SAIL supports a user-configurable time window and abandons data timestamped prior to that window.

Eye witness reports are represented in a timestamped relational way, similarly to the sensor data. As they are originally expressed in a form of controlled natural language (CNL), some preprocessing is needed to arrive at a relational form.

The control program presented in Algorithm 6, coordinates the processing of all that. The control procedure takes two arguments: a TBox and a configurable window size. There are two arrays \( sd \) and \( data \) trace. Each array is of size \( n \) and each element in the array is initially set to an empty set. The main part of the procedure consists of a while-loop that runs indefinitely. In each iteration, the procedure is blocked until the next input arrives from the CNL assertion handler. It then proceeds by reading user queries in nRQL format from the query handler. Positions of array \( sd \) are filled with sensor data in sequence from index \( i \) to \( n - 1 \), as \( t \) increments with each iteration.
Data Aggregation and Semantic Analysis

Semantic Analysis
(Description Logic/Racer)

Data Aggregation
(Rules/E-KRHyper)

GIS

DB

... ABox_j ... ABox_{j+1} ... ABox_{j+2} ...

... SD_i ... SD_{i+1} ... SD_{i+2} ...

Alert generation
(LTL/BA)

CNL generation

ABox

CNL parsing/CNL generation

CNL Answer
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, surfectriker
  - 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).
Some More Features

- Preserve information over time

  \[\text{reassert(take\_off(Event, Obj, CreationTime)) : - take\_off(Event, Obj, CreationTime), current\_time(Time), object(Obj, Time).} \%\text{reassert as long as Obj exists}\]

- 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
(Description Logic/Racer)

Data Aggregation
(Rules/E-KRHyper)

ABox

SD

SD

GIS

DB

Eye-witness reports

CNL parsing/CNL generation/CNL Answer

Alert generation
(LTL/BA)

System Architecture

SD

SD

SD

...
Semantic Analysis

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

\[
\text{aggressive} \triangleq \exists \text{ has}\_\text{target}. \\
(\text{physical}\_\text{object} \sqcup \text{space}\_\text{region})
\]

Data Aggregation provides concept/role assertions

\[
\text{has}\_\text{target}(\text{obj1}, \text{obj2}). \\
\text{physical}\_\text{object}(\text{obj2}).
\]

It follows \text{aggressive}(\text{obj1})
Ontology contains

\[ \text{aggressive} \equiv \exists \text{has_target}. \]
\[ (\text{physical_object} \sqcup \text{space_region}) \]

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

\[ \text{aggressive(obj1)}. \]

it follows that obj1 has a target that is a
\[ \text{physical_object} \text{ or a } \text{space_region} \]
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

**System Architecture**

- **Data Aggregation (Rules/E-KRHyper)**
- **Alert generation (LTL/BA)**
- **CNL Query**
- **CNL Answer**
- **CL Alert**

**Processes**

- **CNL parsing/CNL generation**
- **Query interface (nRQL)**
- **Alert generation (LTL/BA)**
- **Semantic Analysis (Description Logic/Racer)**

**Components**

- **ABox**
- **DB**
Alert Generation

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

G(¬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.''

\[ \varphi := G(in\_air(p) \Rightarrow \neg cross\_border(p) \cup landed(p)). \]

- Caveat: no reasoners for temporal description logics available
Alert Generation by Runtime Verification

- Operationalization of a LTL formula $\phi$ 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 $uw' \not\models \phi$

\[
\phi := G(in\_air(p) \Rightarrow \neg cross\_border(p) U landed(p)).
\]
Alerts via Description Logic

FSM monitor: state $S$ and predecessors

For each state $S$

For individuals that satisfy LHS query, populate concept $C_s$ in the next time point.
Example: \( \varphi := G(\text{in\_air}(p) \Rightarrow \neg \text{cross\_border}(p) \cup \text{landed}(p)) \).

Corresponding concepts/queries for three states:

\[
C_1 : (C_1 \cap \neg \text{in\_air}) \cup (C_2 \cap \text{landed}), \\
C_2 : (C_2 \cap \neg \text{landed} \cap \neg \text{cross\_border}) \cup (C_1 \cap \text{in\_air}), \\
C_3 : C_3 \cup (C_1 \cap \text{cross\_border}) \cup (C_2 \cap \text{cross\_border}).
\]
Controlled Natural Language

- CNL Query
- CNL Answer
- CL Alert

CNL parsing
- CNL generation

Query interface
- (nRQL)

Semantic Analysis
- (Description Logic/Racer)

Alert generation
- (LTL/BA)

CNL generation
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
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