A semantic approach for identifying assurance deficits in unmanned aerial vehicle software

A. Groza, I. A. Letia, A. Goron, S. Zaporojan

Department of Computer Science - Technical University of Cluj-Napoca

August 13, 2014
Assuring safety in complex technical systems is a crucial issue in several critical applications like air traffic control or medical devices.

Safety assurance and compliance to safety standards may prove to be a real challenge when we deal with adaptive systems.

Argument-based safety cases offer a plausible alternative basis for certification of critical software.
Objectives

- Create a framework for evidential reasoning with visualization support
- Apply an efficient validation method for competing hypotheses
The Challenge

Formalize the GSN graphical notation in DL such as to permit automatic reasoning on each diagram of the safety cases.

Limitations:

- Transform GSN notation to DL notation
- Find an appropriate declarative language which allows reasoning
The Approach

- Provide a formal representation of the argumentative-based Goal Structuring Notation (GSN) standard
The Approach

- Provide a formal representation of the argumentative-based Goal Structuring Notation (GSN) standard
- Exploit reasoning in description logic to identify assurance deficits in the GSN model
The Approach

- Provide a formal representation of the argumentative-based Goal Structuring Notation (GSN) standard
- Exploit reasoning in description logic to identify assurance deficits in the GSN model
- Flaws are given to a hybrid logic-based model checker to be validated against a Kripke model
The solution is based on three technical instrumentations:
- the $SHI$ version of DL
- the GSN standard
- hybrid logics (HLs)
Example

- An Unmanned Aircraft System consists of:
  - UAV itself: equipped with an autonomous control system
  - ground station
  - Air Traffic Management: provides the required coordinates for the UAV

- Goal: prove that an UAV can complete safely its mission and that all the major implied risks are mitigated
A semantic approach for identifying assurance deficits in unmanned aerial vehicles
<table>
<thead>
<tr>
<th>State</th>
<th>Parameters</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>PathFollowing</td>
<td>(\neg obs)</td>
<td>UAV follows the path on the given corridor</td>
</tr>
<tr>
<td>DetectObstacles</td>
<td>(obs)</td>
<td>Obstacles are signaled by sensors</td>
</tr>
<tr>
<td>AvoidObstacles</td>
<td>(obs \land d)</td>
<td>UAV performs an avoidance maneuver</td>
</tr>
<tr>
<td>FaultControl</td>
<td>(errObs \lor errAvoid)</td>
<td>Error signaled by Detect or Avoid</td>
</tr>
<tr>
<td>Land</td>
<td>(\neg obs \lor errObs \lor errAvoid)</td>
<td>UAV performs the landing procedure</td>
</tr>
</tbody>
</table>

\[ \text{PathFollowing} \rightarrow \text{DetectObstacles} \rightarrow \text{AvoidObstacles} \rightarrow \text{FaultControl} \rightarrow \text{Land} \]
## Modeling the Goal Structuring Notation in DL

Retrieving information about the GSN model

<table>
<thead>
<tr>
<th>Query</th>
<th>RacerPro query</th>
<th>RacerPro answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top level goal</td>
<td>(concept – instances TopLevelGoal)</td>
<td>$g_1$</td>
</tr>
<tr>
<td>Support goals</td>
<td>(concept – instances SupportGoal)</td>
<td>$g_2, g_3, g_4, g_5$</td>
</tr>
<tr>
<td>Evidence supporting goal $g_1$</td>
<td>(individual – fillers $g_1$ hasEvidence)</td>
<td>$e_1, e_2$</td>
</tr>
<tr>
<td>Evidence verified against the model $m_1$</td>
<td>(individual – fillers $m_1$ (inverse hasModel)))</td>
<td>$e_1, e_2, e_3, e_4, e_5$</td>
</tr>
<tr>
<td>Evidence not verified</td>
<td>(concept – instances (and Evidence (some hasTestResult False)))</td>
<td>$e_1, e_2, e_3, e_4, e_5$</td>
</tr>
<tr>
<td>Goals supported by not verified evidence</td>
<td>(concept – instances NotVerifiedGoals $g_1, g_2, g_3, g_4, g_5$)</td>
<td></td>
</tr>
</tbody>
</table>
Interleaving reasoning with HL and DL for identifying assurance deficits

Our method interleaves two steps:

- Check with hybrid logic if the evidence nodes from the GSN representation have their corresponding formulas validated against the Kripke model.
- By reasoning in DL, we identify which goals in the GSN model are not supported by verified evidence.
The verification uses the following parameters:

- the minimum distance $d_{min}$ allowed between the UAV and another object without risk of collision;
- the reported coordinates $c_{uav}$ by the UAV; and
- the given coordinates $c_{ATM}$ by the ATM.
- the reported distance $d_{obs}$ between the UAV and another approaching UAV
- the minimum distance $d_{min}$ allowed without any risk of collision
For evidence $e_1$:

$$f_1 = \Downarrow i(c_{ATM}) \rightarrow i[F](c_{ATM} > d_{\text{min}})$$  \hspace{1cm} (1)

For evidence $e_2$:

$$f_2 = \Downarrow i(c_{ATM}) \rightarrow @t_i[\text{Next}](c_{\text{uav}} = c_{ATM})$$  \hspace{1cm} (2)

The justification $j_2$ of the sub-goal $g_2$ supported by $e_1$ and $e_2$ is expressed as:

$$j_2 = \Downarrow i(c_{\text{uav}} = c_{ATM}) \rightarrow @i[F](c_{\text{uav}} > d_{\text{min}})$$  \hspace{1cm} (3)

The implication $f_1 \land f_2 \rightarrow j_2$ is true (the following of the coordinates from the ATM ensures the required min safe distance).

If we bind to $i$ the state in which the reported distance between the UAV and another approaching UAV is less than the min one then for all future states the reported distance must be kept higher than 0.

If we bind to nominal $i$ the state in which an obstacle is signaled by the sensors, then the reported distance to the obstacle must be maintained different than 0 until it becomes higher than the min established threshold.

Evidence $e_3$ is formally expressed as:

$$f_3 = \Downarrow i(d_{\text{obs}} < d_{\text{min}}) \land @i(d_{\text{obs}} < d_{\text{min}}) \rightarrow \Downarrow i(\text{obs})$$  \hspace{1cm} (5)

Evidence $e_4$ is formally expressed as:

$$f_4 = \Downarrow i(\text{obs}) \rightarrow @i((d_{\text{obs}} \neq 0) \cup (d_{\text{obs}} > d_{\text{min}}))$$  \hspace{1cm} (6)

Evidences $e_3$ and $e_4$ are used to validate the sub-goal $g_4$.

To complete the validation of $g_4$, we have to prove the formula $f_3 \land f_4 \rightarrow j_4$, which is true (the presence of an obstacle indicated by an observed distance, which is less than the min accepted one will entail an avoidance maneuver).
Updating the Abox for the GSN model with the newly validated evidences:

\[
\begin{align*}
(e_1, f_1) & : \text{hasFormula}, & (e_1, "true") & : \text{hasTestResult} \\
(e_2, f_2) & : \text{hasFormula}, & (e_2, "true") & : \text{hasTestResult} \\
(g_2, j_2) & : \text{hasJustification}, & (f - g_2, "true") & : \text{hasTestResult} \\
(e_3, f_3) & : \text{hasFormula}, & (e_3, "true") & : \text{hasTestResult} \\
(e_4, f_4) & : \text{hasFormula}, & (e_4, "true") & : \text{hasTestResult} \\
(g_4, j_4) & : \text{hasJustification}, & (f - g_4, "true") & : \text{hasTestResult}
\end{align*}
\]
Conclusions

- Combining argumentation and model checking might bring about additional advantages such as preliminary validation of argumentation schemes constructed to support safety cases,

- Ensures that the stability of the system will not be affected by the available choices

- Foresees possible impediments in selecting one option over another

- Abstractization was used to complement the more visually used GSN standard with a formalized model

- This joint approach will increase the degree of trust in certifying the correct functioning of critical safety systems.
Contributions:

- Integrate hybrid logic with argumentation theory
- Provide a formal model of the GSN standard in description logic
- GSN structures safety cases, HL is able to validate evidence nodes
- DL provides a middleware language to integrate GSN and model checking
- DL was used to analyze the status of the arguments and their supporting evidence
- A step towards a formal model for the GSN standard
- Current work is focused on investigating the feasibility of the solution against large-scale safety cases
Thank you!