KU LEUVEN

vGOAL: a GOAL-based Specification Language for Safe Autonomous Decision-Making

DistriNet Research Group
Authors: Yi Yang, Tom Holvoet
Presenter: Yi Yang

Motivating Example

- Safe Autonomous logistic system
  - Safe (high-level) decision-making
  - Safe (low-level) code execution
  - Safe computing hardware

Research Scope

- Autonomous System
  - Agent-based model
  - Cooperative to achieve goals
  - Competing for critical resources
- Safe autonomous decision-making[2]
  - Avoids deliberately pursuing unsafe behaviors based on its beliefs and goals.
- Safe autonomous decision-making component: [1]
  - High-level discrete controller
  - Work independently and closely with low-level continuous controller

1. Louise A Dennis, Michael Fisher, Nicholas K Lincoln, Alexei Lisitsa, and Sandor M Veres. 2016. Practical verification of decision-making in agent-based autonomous systems. Automated Software Engineering 23 (2016), 305–359.
2. Louise Dennis and Michael Fisher. 2021. Verifiable autonomy and responsible robotics. Software Engineering for Robotics (2021), 189–217.

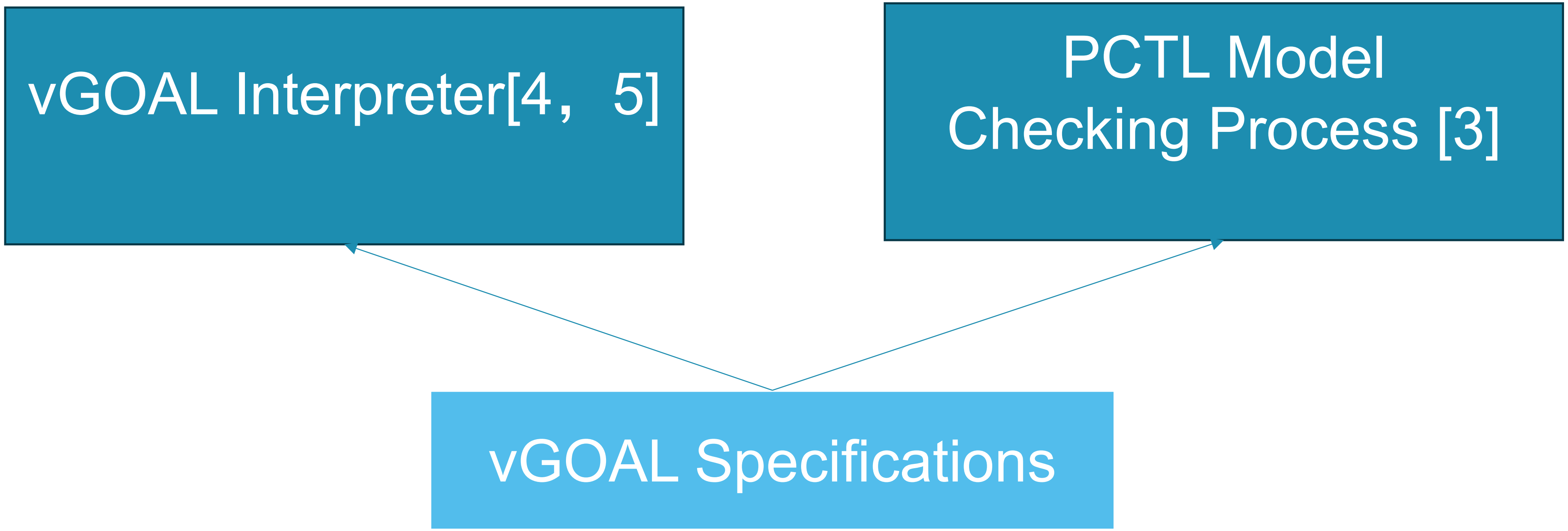
Safe Autonomous Decision-Making

vGOAL Interpreter[4， 5]

PCTL Model Checking Process [3]

3. Yi Yang, Tom Holvoet, Making Model Checking Feasible for GOAL, 10th International Workshop on Engineering Multi-Agent Systems, 10th International Workshop on Engineering Multi-Agent Systems, Auckland, New Zealand (Online), May 9-10, 2022
4. Yi Yang, Tom Holvoet, Generating Safe Autonomous Decision-making in ROS, Fourth Workshop on Formal Methods for Autonomous Systems, Proceedings FMAS2022 ASYDE2022, volume 371, pages 184-192, Berlin, Germany, September 26-27, 2022
5. Yang, Y., Holvoet, T.: Safe autonomous decision-making with vGOAL. In: Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection. Springer (2023)

Safe Autonomous Decision-Making



3. Yi Yang, Tom Holvoet, Making Model Checking Feasible for GOAL, 10th International Workshop on Engineering Multi-Agent Systems, 10th International Workshop on Engineering Multi-Agent Systems, Auckland, New Zealand (Online), May 9-10, 2022
4. Yi Yang, Tom Holvoet, Generating Safe Autonomous Decision-making in ROS, Fourth Workshop on Formal Methods for Autonomous Systems, Proceedings FMAS2022 ASYDE2022, volume 371, pages 184-192, Berlin, Germany, September 26-27, 2022
5. Yang, Y., Holvoet, T.: Safe autonomous decision-making with vGOAL. In: Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection. Springer (2023)

Why a new specification language?

- Safe-by-generation decisions
  - no extra formal verification to ensure safety
- A purely logical approach
  - No hard-encoded component for error handling
- Easy integration with ROS
  - More applicable to the existing ROS-based applications

vGOAL Specification Language

- A purely logical language (first-order logic)
  - Choose GOAL as the basis
    - Expressive for specifying autonomous decision-making
    - Logic-driven decision-making generation mechanisms
    - Suitable for formal verification
  - Verifiable GOAL
    - Safe-by-generation decisions
    - Automated PCTL model checking process

- Automated reasoning
  - No negative recursion
  - A finite domain of each variable
  - Quantified variables

vGOAL Specifications

```
goal_base3 = ['delivered(2,3)']
goal_base4 = ["delivered(2,4)"]
goals3 = [goal_base3, goal_base4]
```

```
safety = {"A1": ["safe1","safe2"], "A2": ["safe1","safe2"], "A3": ["safe1","safe2"]}
A1 = DG.Agent("A1", belief_base1, goals1)
A2 = DG.Agent("A2", belief_base2, goals2)
A3 = DG.Agent("A3", belief_base3, goals3)
C = DG.Agent("C", belief_base4, goals4)
Agents = [A1,A2,A3,C]
```

```
"exists l. battery(l) and equal(l,1) implies safe1",
"exists l. battery(l) and equal(l,2) implies safe1",
"exists p. at(p) and not at(9) implies safe2",
```

vGOAL Specifications

```
"E1 implies nonfatal",
"E2 implies nonfatal",
"E3 implies nonfatal",
"E4 implies fatal",
```

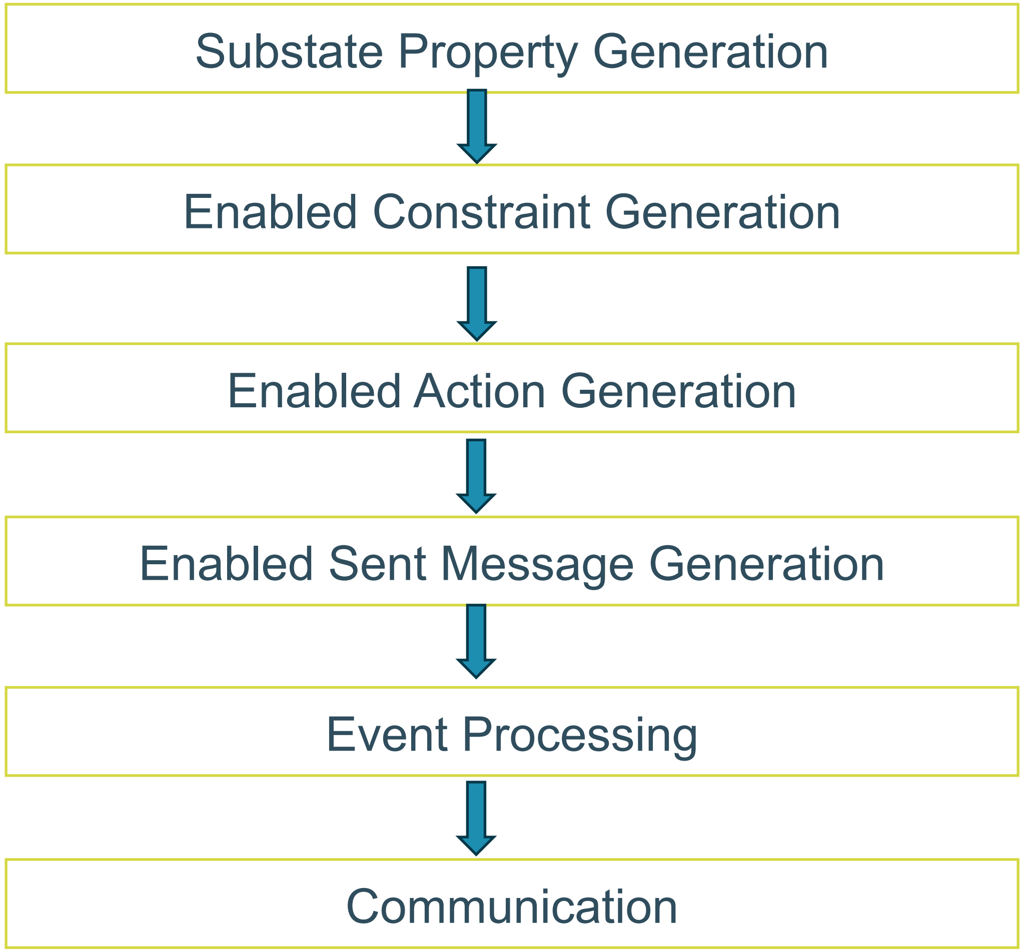
```
"fatal implies drop all",
"fatal implies delete all",
"nonfatal and not goal_change implies drop all",
"nonfatal and not goal_change implies adopt located(charging)",
"nonfatal and not goal_change implies adopt at(5)",
"nonfatal and not goal_change implies insert goal_change",
"nonfatal and E1 implies delete E1",
"nonfatal and E2 implies delete E2",
"nonfatal and E3 implies delete E3",
```

Formal Syntax

$$
\begin{aligned}
id &::= string \\
b &::= ground\_atom \\
g &::= ground\_atom \\
B_{sensor} &::= B_{sensor} \cup \{b\} | \emptyset \\
B_{prior} &::= B_{prior} \cup \{b\} | \emptyset \\
B &::= B_{sensor} \cup B_{prior} \\
G &::= G \cup \{g\} | \emptyset \\
goals &::= G : goals | [\,] \\
p &::= predicate \\
neg\_p &::= \neg p \\
R &::= all | allother | id \\
msg_s &::= send{:}(R,p) | send!(R,p) | send?(R,p) \\
msg_r &::= sent{:}(R,p) | sent!(R,p) | sent?(R,p) \\
M_S &::= M_S \cup \{msg_s\} | \emptyset \\
M_R &::= M_R \cup \{msg_r\} | \emptyset \\
Agent &::= (id, B, goals, M_S, M_R) \\
MAS &::= MAS \cup \{Agent\} | \emptyset
\end{aligned}
$$

$$
\begin{aligned}
D &::= D \cup \{constant\} | \emptyset \\
hs &::= hs \wedge p | hs \wedge neg\_p | True \\
rule_1 &::= hs \rightarrow p \\
qrule_1 &::= \forall x.qrule_1 | \forall x \in D.qrule_1 | \exists x.qrule_1 | rule_1 \\
K &::= K \cup \{qrule_1\} | K \cup \{ground\_atom\} | \emptyset \\
lh &::= a\text{-}goal(p) \wedge hs \\
rule_2 &::= lh \rightarrow p \\
qrule_2 &::= \forall x.qrule_2 | \forall x \in D.qrule_2 | \exists x.qrule_2 | rule_2 \\
C &::= C \cup \{qrule_2\} | \emptyset \\
A &::= A \cup \{qrule_1\} | \emptyset \\
rule_3 &::= hs \rightarrow hs \\
qrule_3 &::= \forall x.qrule_3 | \forall x \in D.qrule_3 | \exists x.qrule_3 | rule_3 \\
E &::= E \cup \{qrule_3\} | \emptyset \\
rule_4 &::= hs \rightarrow msg_s \\
qrule_4 &::= \forall x.qrule_4 | \forall x \in D.qrule_4 | \exists x.qrule_4 | rule_4 \\
S &::= S \cup \{qrule_4\} | \emptyset \\
update &::= insert(b) | delete(b) | adopt(g) | drop(g) \\
response &::= msg_s | update \\
rule_5 &::= msg_r \wedge hs \rightarrow response \\
qrule_5 &::= \forall x.qrule_5 | \forall x \in D.qrule_5 | \exists x.qrule_5 | rule_5 \\
rule_6 &::= lh \rightarrow response \\
qrule_6 &::= \forall x.qrule_6 | \forall x \in D.qrule_6 | \exists x.qrule_6 | rule_6 \\
rule_7 &::= hs \rightarrow response \\
qrule_7 &::= \forall x.qrule_7 | \forall x \in D.qrule_7 | \exists x.qrule_7 | rule_7 \\
P &::= P \cup \{qrule_5\} | P \cup \{qrule_6\} | P \cup \{qrule_7\} | \emptyset
\end{aligned}
$$

Operational Semantics



Logical Derivation and Minimal Model Generation

$$EP ::= model_C \cup M_R \cup P,$$

$$PR ::= \{\rho(response) | EP \models \rho(response) \wedge model_C \cup P \not\models \rho(response)\}.$$

Functions

$$update(B) ::= B \cup \bigcup^{m} \rho(\{B_m\}) \setminus \bigcup^{n} \rho(\{B_n)\},$$

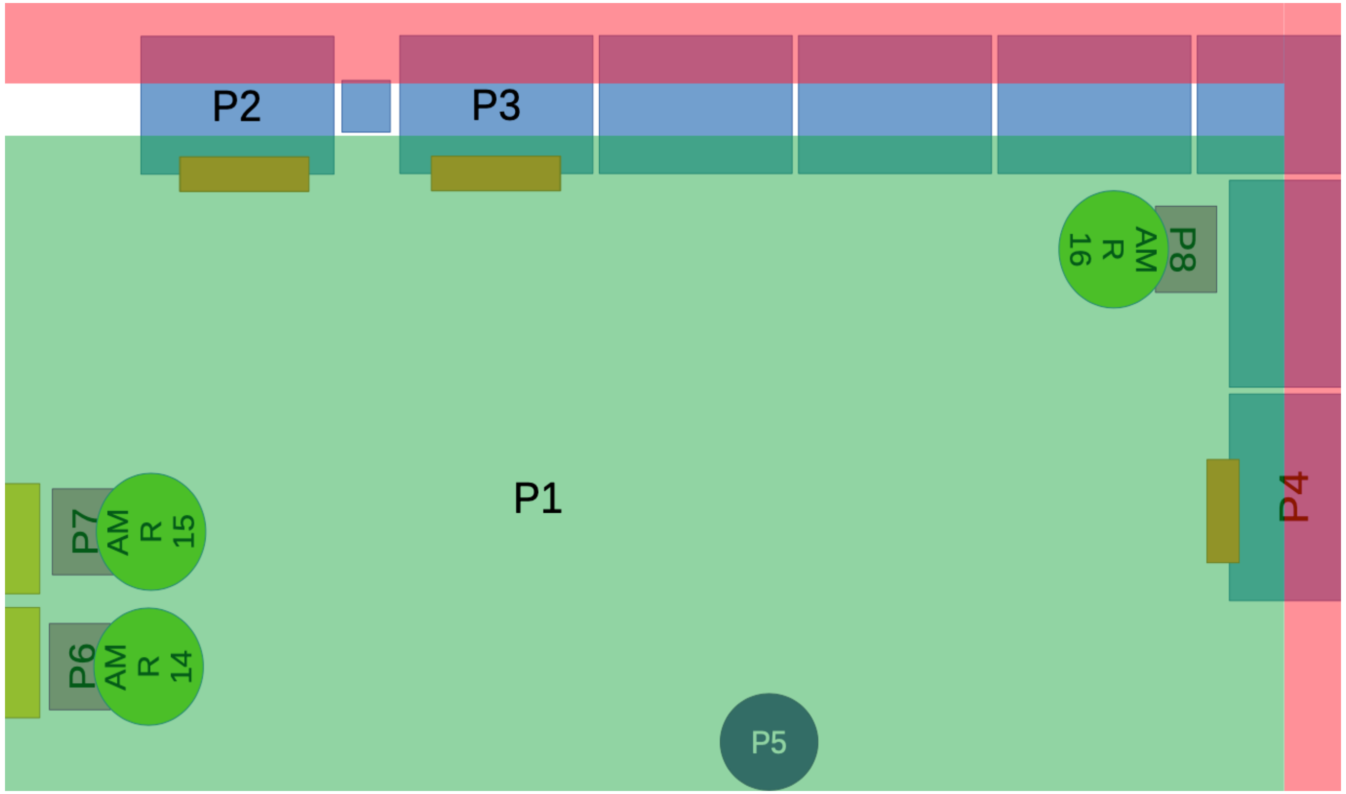
$$T(substate, GE) ::= \begin{cases} id : (update(B), goals), \text{ if } GE = \{\rho(hs)\}, \\ id : (B, goals), \text{ if } GE = \emptyset, \end{cases}$$

$$substate ::= T(substate, GE).$$

KU LEUVEN

Case Study: Autonomous Logistic System

- Task
  - Three autonomous mobile robots deliver four workpieces from one of the picking-up stations (P3 and P4) to the delivery station (P2).
- Requirements
  - Autonomous decision-making generation
  - Real-time information processing
  - Error-handling
  - Goal redistribution
  - Competing requests resolution
  - Safety assurance



Case Study: Autonomous Mobile Robots

- Initial Goal Distribution
  - Agent 1: 1 delivery goal
  - Agent 2: 1 delivery goal
  - Agent 3: 2 delivery goal
- Error Handling
  - Non-fatal error: abandon the current delivery goal, and go to P5.
  - Fatal error: redistribute the remaining goals of the agent to other active agents.
- Safety Requirements
  - Safe battery level
  - Safe location
- Demo Video:
  - Error-free run: Demo_No_Error_Run.mp4
  - A run including a non-fatal error: Demo_including_a_non_fatal_errror.mp4
  - A run including a fatal error: Demo_including_a_fatal_error.mp4

Efficiency (vGOAL Interpreter)[5]

- Experiments
  - over 100 runs
  - Each run: 6-8 min
  - Sensor updates: every 0.5s
  - Total decisions: 72000-96000

Repeated	Active Agent	Decision	Error	Safety Checking(s)	Execution Time(s)
Yes	-	-	-	0	4.28E-5
No	3	2	No	2.10E-6	0.82
No	3	1	No	1.02E-6	0.64
No	3	0	No	0	0.69
No	2	1	No	1.02E-6	0.49
No	2	0	No	0	0.48
No	1	1	No	1.02E-6	0.38
No	1	1	Fatal	1.02E-6	0.36
No	1	1	Non-Fatal	1.02E-6	0.41
No	1	0	No	0	0.35

- Phenomena
  - The sensor updates are mostly repeated compared with the last sensor information.
  - In most cases, 0 or 1 decision is generated.

- Observation
  - Handles repeated information quickly
  - Execution time increases almost linearly with the number of active agents.
  - Not much time difference to generate 0 or 1 decision.
  - Time cost for safety checking is little.
  - The specification order affects execution time.

5. Yang, Y., Holvoet, T.: Safe autonomous decision-making with vGOAL. In: Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection. Springer (2023)
Note: The information on this slide is sourced from [5].

Discussion

- Compared with GOAL, Gwendolen, and AgentSpeak (Jason)
  - No extra formal verification process for safety checking
  - Little additional computation for safety checking.
  - No hard-encoded component for error handling.
  - Python implementation (easy integration to ROS).
  - The least performatives in the communication.
  - Efficiency?

Future Work

- Correctness of the interpreter for vGOAL (program verification, partly proved using SAT solver)

- Investigate how to integrate safe reinforcement learning (safe shielding) with the vGOAL interpreter, focusing on safe motion planning.

- Empirical analysis with GOAL, Gwendolen, and AgentSpeak (Jason)

Reference

1. Louise A Dennis, Michael Fisher, Nicholas K Lincoln, Alexei Lisitsa, and Sandor M Veres. 2016. Practical verification of decision-making in agent-based autonomous systems. Automated Software Engineering 23 (2016), 305–359.
2. Louise Dennis and Michael Fisher. 2021. Verifiable autonomy and responsible robotics. Software Engineering for Robotics (2021), 189–217.
3. Yi Yang, Tom Holvoet, Making Model Checking Feasible for GOAL, 10th International Workshop on Engineering Multi-Agent Systems, 10th International Workshop on Engineering Multi-Agent Systems, Auckland, New Zealand (Online), May 9-10, 2022
4. Yi Yang, Tom Holvoet, Generating Safe Autonomous Decision-making in ROS, Fourth Workshop on Formal Methods for Autonomous Systems, Proceedings FMAS2022 ASYDE2022, volume 371, pages 184-192, Berlin, Germany, September 26-27, 2022
5. Yang, Y., Holvoet, T.: Safe autonomous decision-making with vGOAL. In: Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection. Springer (2023)