

Safety and Certification in Increasingly Autonomous Aviation Systems

Natasha Neogi

Invited Talk at Workshop on Safe and Reliable Robot Autonomy under Uncertainty

International Conference on Robotics and Automation

May 27, 2022

Outline

- Introduction and Motivation
- Urban Air Mobility Concept and Issues
- Verification and Validation Approach
- Case Study
- Conclusions and Future Work

*Unless otherwise noted, all figures, images, photos, etc. were created by the author using LaTEX or Microsoft products



Motivation and Problem Statement

- Increasingly Autonomous Systems are collaborating or coordinating with humans in order to execute safety critical missions
- Autonomous agents need to be rigorously analyzed and assured to build trust
  - Definition and Satisfaction of requirements
  - Correctness
  - Boundaries of performance

Problem Statement

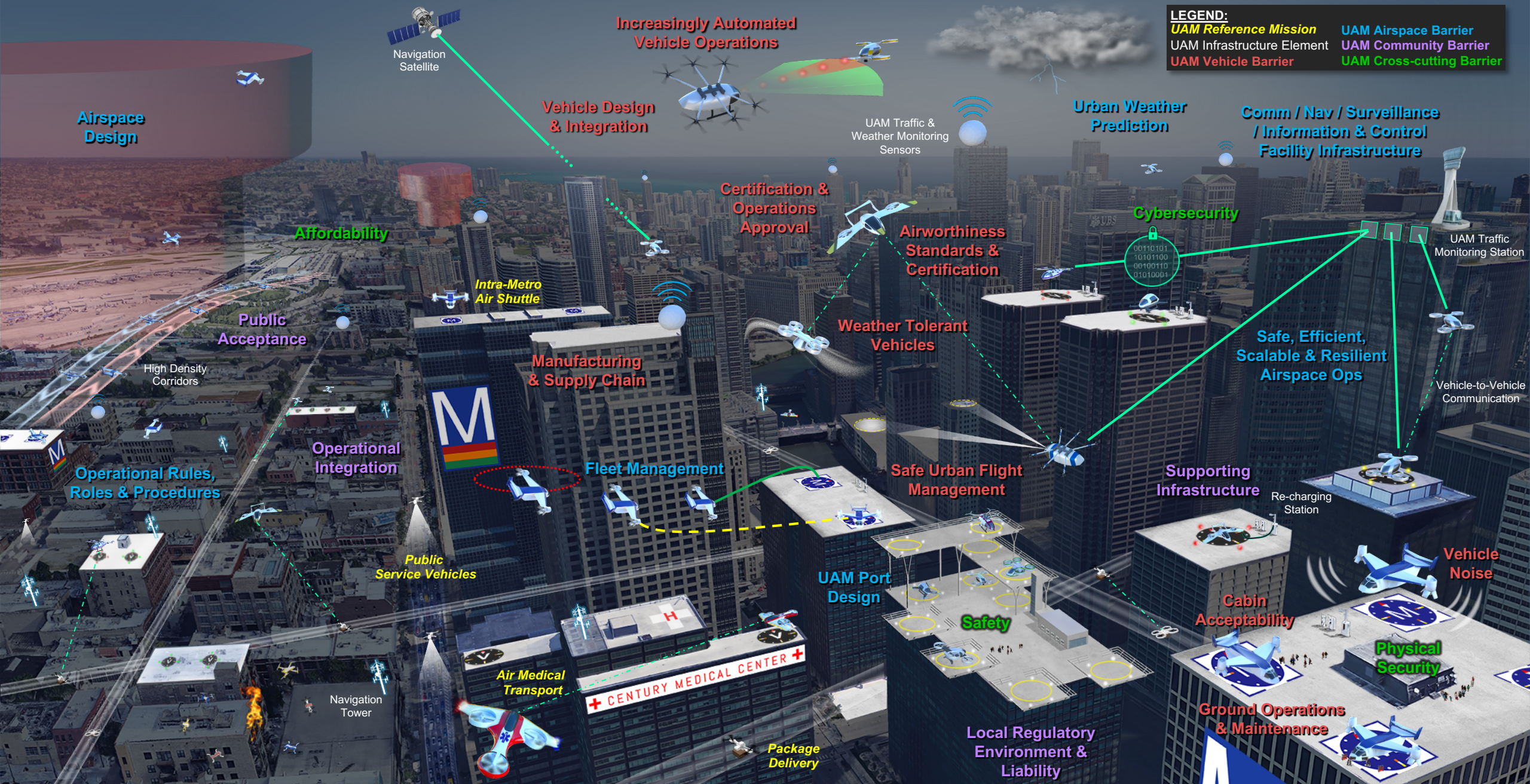
- Is it possible to provide formal guarantees on the behavior of learning systems, specifically with respect to safety properties?

Impact

- Enabling use of learning systems in high confidence settings
  - Verification & Validation
  - Certification (Safety cases & Proof Artifacts)

Urban Air Mobility: Passenger Carrying OV-1

National Aeronautics and Space Administration



Outline

- Introduction and Motivation
- Urban Air Mobility Concept and Issues
- Verification and Validation Approach
- Case Study
- Conclusions and Future Work

UAM Reference Missions

Passenger Carrying Reference Missions


E.G., AIR MEDICAL TRANSPORT


E.G., INTRA-METRO AIR SHUTTLE


E.G., UBIQUITOUS INTRA-METRO TAXI

INITIAL STATE **INTERMEDIATE STATE** **MATURE STATE**

Non-Passenger Carrying Reference Missions


E.G., PUBLIC SERVICE VEHICLES


E.G., SMALL PACKAGE DELIVERY


E.G., UAS MULTI-PACKAGE DELIVERY

UAM Passenger Carrying Market Comparisons

Comparisons of Passengers and Numbers of Vehicles per City

Crown Air Metro	2028	2030
# pax trips/day/city	**23,744**	**136,986**
# vehicles/city	**273**	**1,533**
# flights/day/city	5,936	34,247
# pax/flight (avg)	4	4

Aggressive on assumptions (i.e. autonomy) and implementation (i.e. vertiport infrastructure)

BAH (baseline)	Near-mid
# pax trips/day/city	**8,200**
# vehicles/city	**410**
# flights/day/city	5,500
# pax/flight (avg)	1.49

Fairly conservative assumptions in all except manufacturing volumes, and no predictions for dates

Uber Air Taxi	2025-2030
# pax trips/day/city	**60,000**
# vehicles/city	**400**
# flights/day/city	15,000
# pax/flight (avg)	4

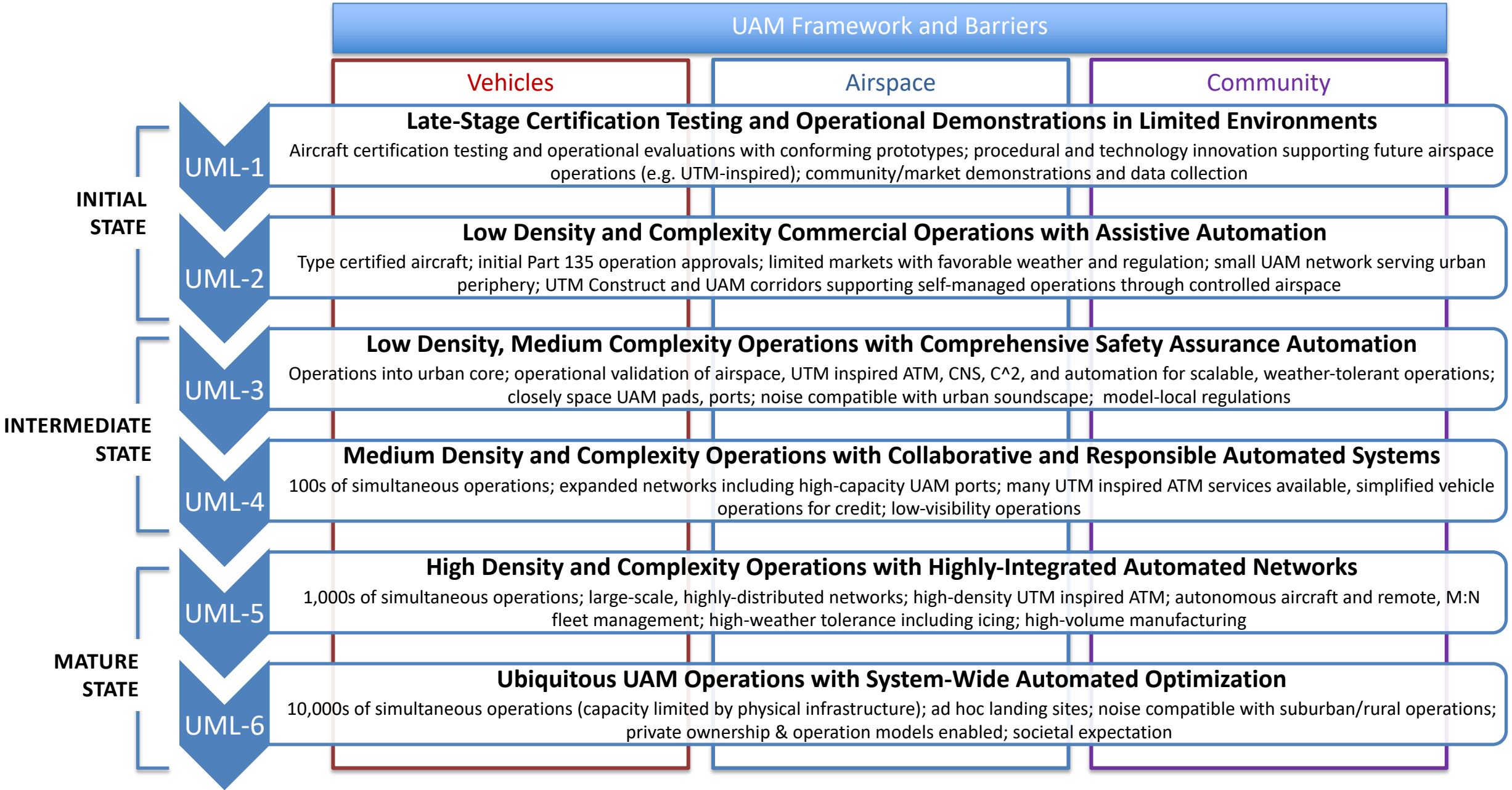
Aggressive timeline and assumptions more in-line with Crown market study

Cost Comparisons Over Time ($/pax-mile)

Cost ($/pax-mile)	Launch	"Near-Mid-Term"	"Far-Mid-Term"	"Long Term"
BAH Baseline	N/A	6.25	~2.50	N/A
BAH Low End	N/A	~3.75	~1.50	N/A
Crown	N/A	2.00	1.20	N/A
Uber	5.73	N/A	1.84	0.44

Leverage data to architect a comprehensive approach that defines a realistic progression to open the UAM passenger-carrying market

UAM Maturity Levels (UML)

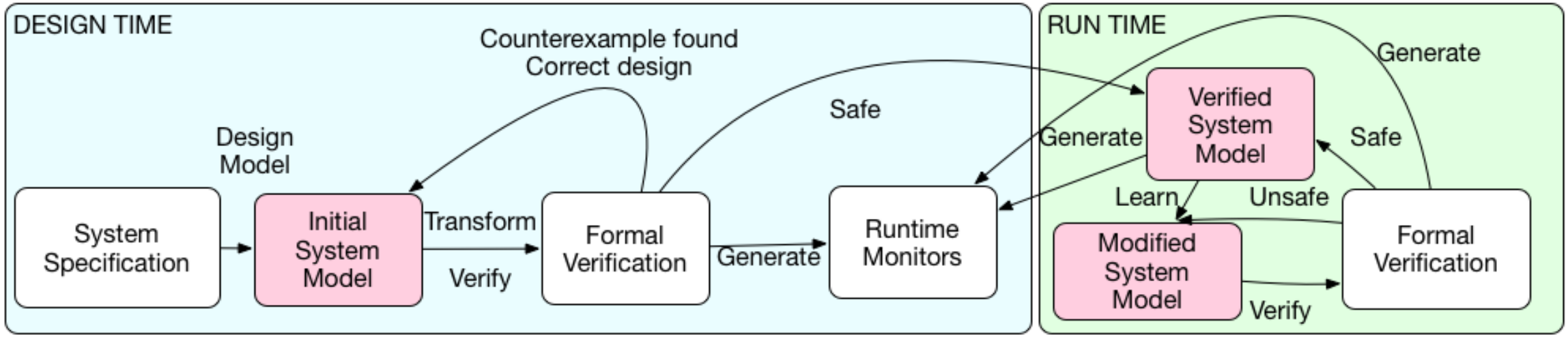


Outline

- Introduction and Motivation
- Urban Air Mobility Concept and Issues
- Verification and Validation Approach
- Case Study
- Conclusions and Future Work

Cognitive Model with Formal Verification Flow

- Agent requirements and constraints are specified in (cognitive) architecture that enables multiple learning mechanisms
- Agent is transformed into formal environment for verification
  - Generates runtime monitors
  - Corrects the present design
- Agent can learn efficient ways
  - Creates or modifies rules which are evaluated and/or verified



Soar Cognitive Architecture [LBA17]

- Agent architecture
  - Integration of several components
    - Perception
    - Memory
    - Production systems (decision procedures)
  - Soar, ACT-R IMPRINT, Air-MIDAS etc.
- Storage/Access/Removal of Working Memory Elements (WMEs) and handling of variables in memory
  - `Identifier ^attribute value`
  - Value is constant or variable
- Soar's Execution Cycle includes substates and impasses
  - Substates created and removed at run time, dynamic
- Learning: chunking and reinforcement learning
  - Chunking: create new rules based on resolution of substates and creates new memory.

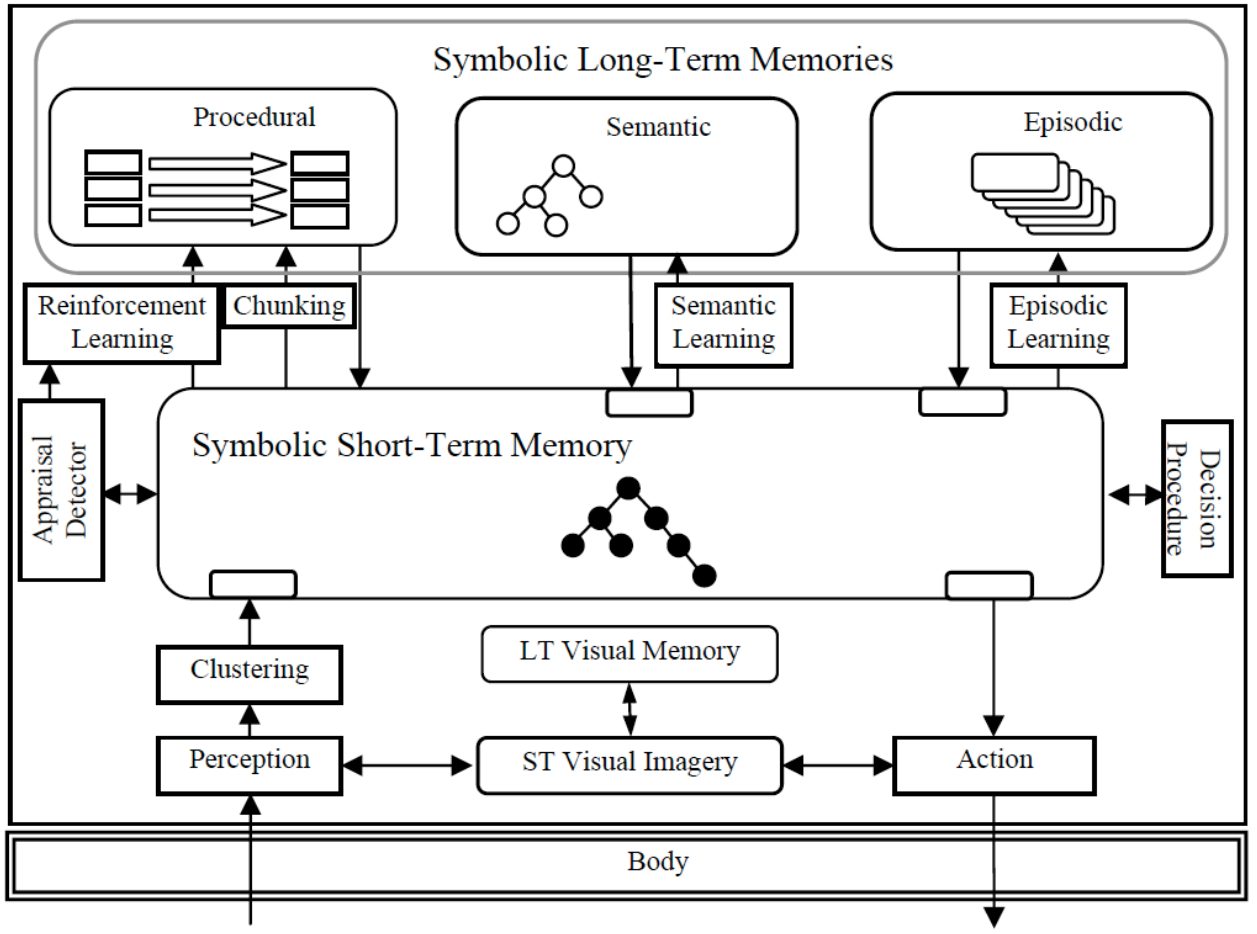
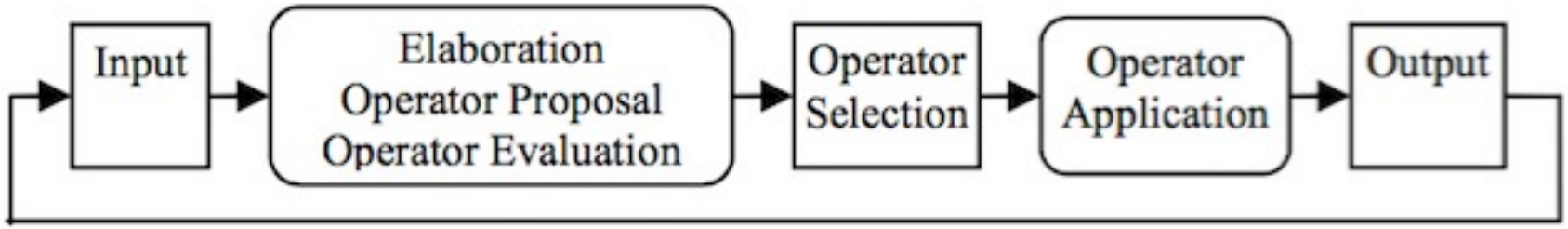


Figure reproduced from [LBA17] Laird, J. E., Bates, C. C., Assanie, M., Derbinsky, N., & Xu, J. (2017). The soar user's manual version 9.6.0. (). Retrieved from https://soar.eecs.umich.edu/downloads/SoarManual.pdf

Soar Processing Cycle [LBA17, LLR06]



Soar processing cycle, Figure reproduced from [LBA17, LLR06]

- Proposed Soar processing cycle
- Generic representation
  - Any rule that is true can be executed
    - Satisfies diverse range of cognitive models

[LBA17] Laird, J. E., Bates, C. C., Assanie, M., Derbinsky, N., & Xu, J. (2017). The soar user's manual version 9.6.0. (). Retrieved from https://soar.eecs.umich.edu/downloads/SoarManual.pdf
[LLR06] Lehman, J., Laird, J., & Rosenbloom, P. (2006). A gentle introduction to soar, an architecture for human cognition: 2006 update., 15. Retrieved from http://web.eecs.umich.edu/~soar/sitemaker/docs/misc/GentleIntroduction-2006.pdf

Timed Automata [AD94]

Definition 1. ***Timed Automaton (TA).*** *A timed automaton is a tuple* (L, l_0, C, A, E, I)*, where* L *is a set of locations;* $l_0 \in L$ *is the initial location;* C *is the set of clocks;* A *is a set of actions, co-actions and the internal* τ*-action;* $E \subseteq L \times A \times B(C) \times 2^C \times L$ *is a set of edges between locations with an action, a guard and a set of clocks to be reset; and* $I : L \to B(C)$ *assigns invariants to locations.*

Definition 2. ***Timed Automaton (TA) Semantics.*** *Let* (L, l_0, C, A, E, I) *be a timed automaton* TA*. The semantics of the TA is defined as a labelled transition system* $\langle S, s_0, \to \rangle$*, where* $S \subseteq L \times \mathbb{R}^C$ *is the set of states,* $s_0 = (l_0, u_0)$ *is the initial state, and* $\to \subseteq S \times \{\mathbb{R}_{\geq 0} \cup A\} \times S$ *is the transition relation such that:*

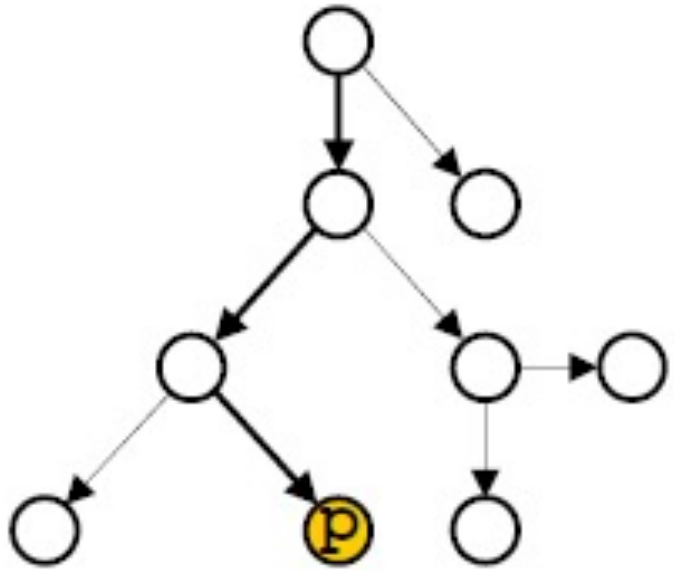
1. $(l, u) \xrightarrow{d} (l, u + d)$ *if* $\forall d' : 0 \leq d' \leq d \Rightarrow u + d' \in I(l)$ *and*
2. $(l, u) \xrightarrow{a} (l', u')$ *if* $\exists e = (l, a, g, r, l') \in E$ *such that* $u \in g, u = [r \mapsto 0]u$ *and* $u' \in I(l)$,

where for $d \in \mathbb{R}_{\geq 0}, u + d$ *maps each clock* x *in* C *to the value* $u(s) + d$*, and* $[r \mapsto 0]u$ *denotes the clock valuation which maps each clock in* r *to* 0 *and agrees with* u *over* $C \setminus r$.

[AD94] Rajeev Alur and David L. Dill. 1994. A theory of timed automata. Theor. Comput. Sci. 126, 2 (April 1994), 183-235. DOI=10.1016/0304-3975(94)90010-8 http://dx.doi.org/10.1016/0304-3975(94)90010-8

Uppaal Model Checking

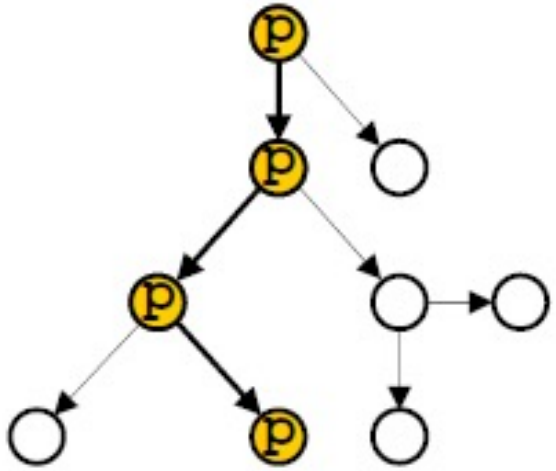
- *A specific condition holds in some state of the model's potential behaviors*
- E◊p "Exists eventually p"
  meaning there is an execution path in which p eventually (in some state of the path) holds



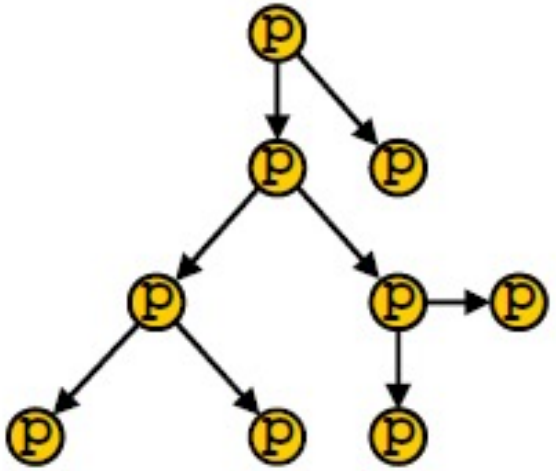
Uppaal Model Checking (contd.)

A specific condition ***holds in all the states*** *of an execution path.* **Two possibilities**:

`E[] p` "Exists globally p" meaning there is an execution path in which p holds for all the states of the path



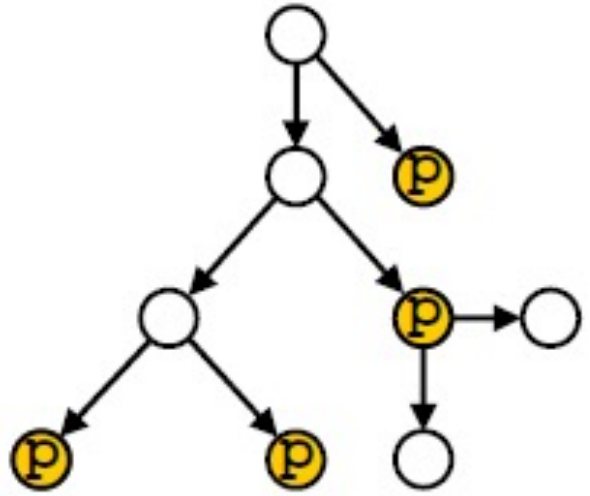
`A[] p` "Always globally p" For each (all) execution path p holds for all the states of the path



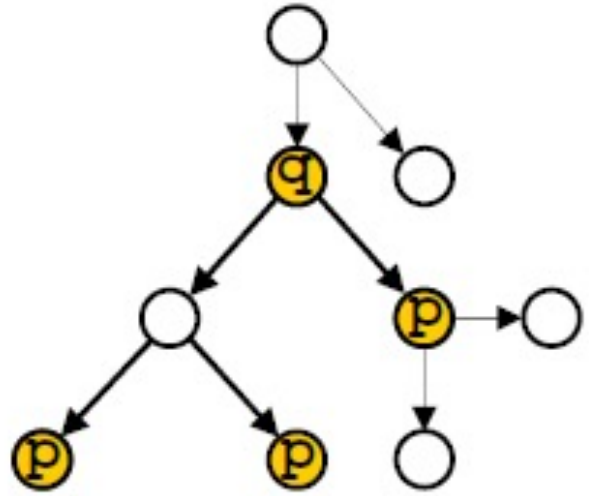
Uppaal Model Checking (contd.)

*A specific condition is **guaranteed to hold eventually** (= at some moment).* **Two possibilities:**

`A<> p` "Always eventually p"
For each (all) execution path p holds for at least one state of the path



`q-->p` "q always leads to p"
Any path that "starts" with a state in which q holds reaches later a state in which p holds



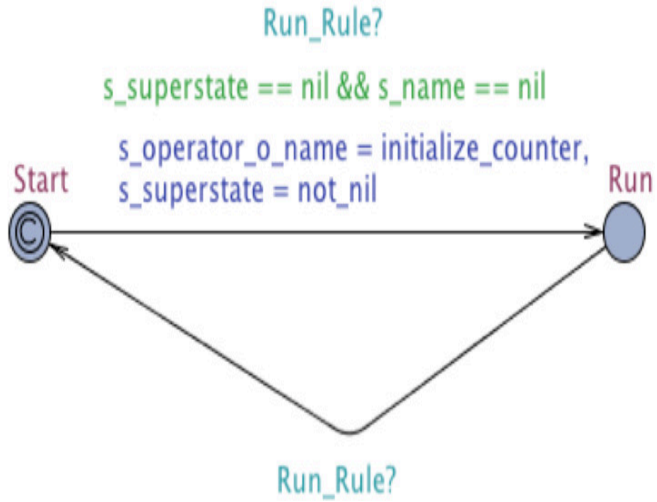
Automated Translation Algorithm (Simplified) [NBG21]

Algorithm 1 Generate $(S, s_0, \rightarrow)$ for $TA = (L, l_0, C, A, E, I)$ from $rname(V, Pre(\{\phi_1, \phi_2, ...\phi_m\}), Post(\{\psi_1, \psi_2, ...\psi_r\}))$

Require: $\forall \phi(v_i), \psi(v_i) \in WFF,$, u: $C \rightarrow \mathbb{R}_{\geq 0}$
Ensure: $fire(rname) \subseteq traces(TA)$
  $l_0 \leftarrow \{Start\}$
  $L \leftarrow$ {Start, Run }
  $s_0 \leftarrow$ (Start, u(x_0))
  $S \leftarrow$ {Start, Run} $\times u(x)$
  I(Start) = {True}
  I(Run) = {True}
  G = { }
  A = {Run_Rule}
  for $j = 1$ to m **do**
    $G \leftarrow G \wedge \phi_j$
  end for
  for $k = 1$ to r **do**
    $A \leftarrow A \wedge \psi_j$
  end for
  if $u \in I$(Start) and L = {Start } **then**
    e_1 = (Start, A, G, $u \mapsto u'$, Run)
  end if
  if $u' \in I$(Run) and L = {Run} **then**
    e_2 = (Run, Run_Rule, {}, $u' \mapsto u''$, Start)
  end if
  $E = \{e_1, e_2\}$
  $S' \xleftarrow{u(x) \cup A} S$

  return $TA = (L, l_0, C, A, E, I)$

```
sp {counter*propose*initialize-counter
   (state <s> ^superstate nil
              -^name)
-->
   (<s> ^operator <o> +)
   (<o> ^name initialize-counter)
}
```

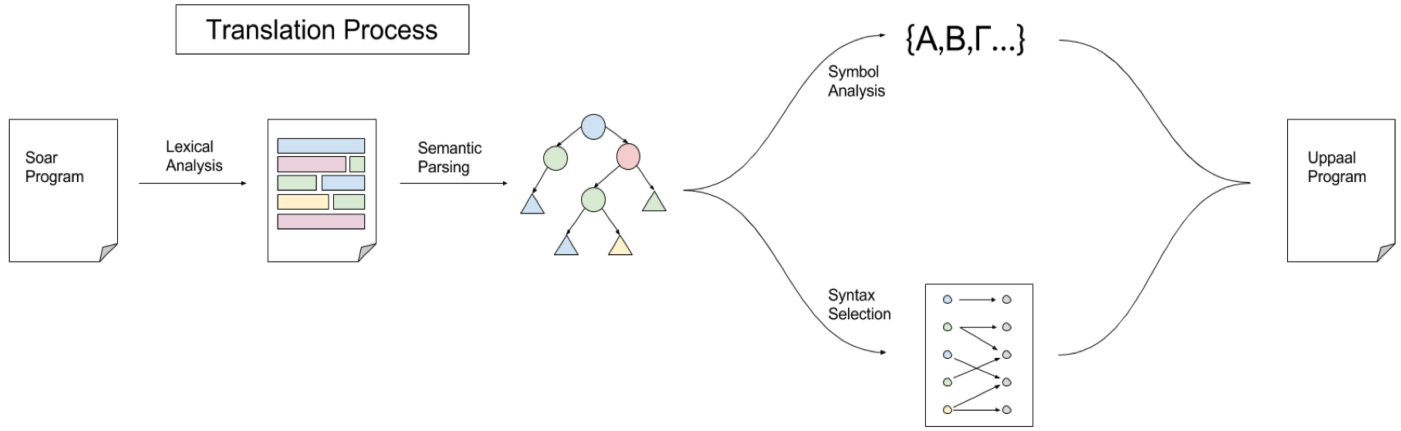


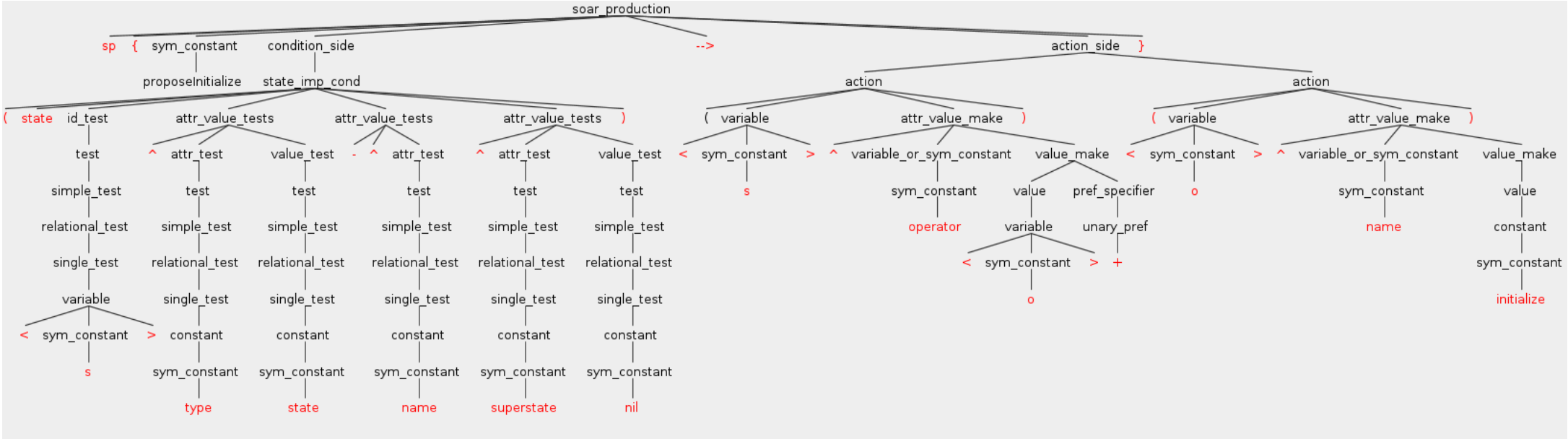
[NBG21] Neogi, N., Bhattacharyya, S., Griessler, D., Kiran, H., and M. Carvalho, "Assuring Intelligent Systems: Contingency Management for UAS," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 6028-6038, Sept. 2021, doi: 10.1109/TITS.2021.3076399.

Implementation of the Translation Algorithm

Challenges in translation:

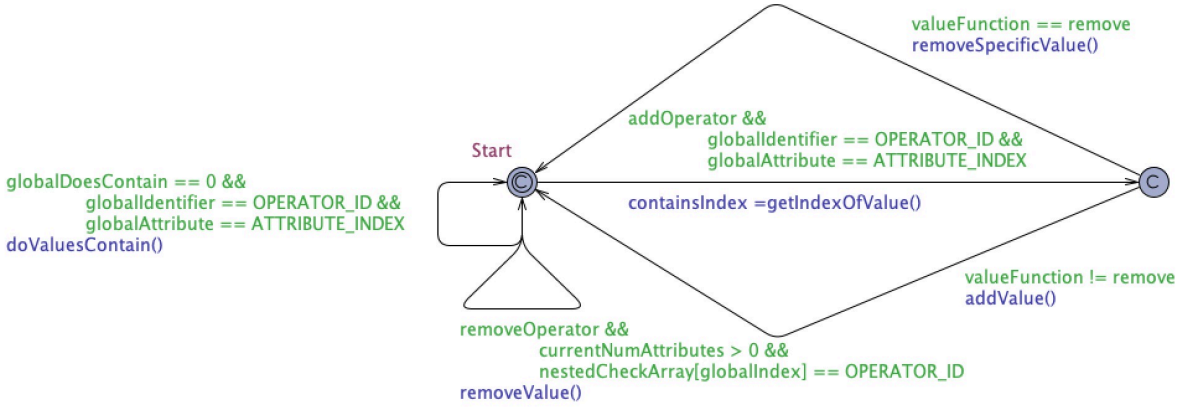
- Understanding architectural interaction
- Production rule execution formalisms
- Cognitive engine flow
- Multi-Valued Attributes
- Preferences





Multi-Valued Attributes and Preferences [NBG21]

- Multi-Valued attributes are stored in array and accessed via Attribute Value Template
  - Single operator enabled for multiple instantiations of variable values
- During Decision Phase, operators are evaluated in a sequence of 8 tests, to select based on preference: (1) Required, (2) Acceptable, (3) Prohibit, (4) Reject, (5) Better-Worse, (6) Best, (7) Worst, and (8) Indifferent.



Algorithm 2 Generate O_p and IT from $PreferenceArray(m,n,p)$ and $O_f = rname(V, V_{bf}, Val_f, Pre(\{\phi_s\}), Post(\{\psi_r\}))$

Require: $\mathbf{card}(O_f) \leq m$, $V_{BF} = \bigcup_{rname(i)} V_{bf}(j)$, $\mathbf{card}(V_{BF} \leq n)$
Require: $\forall v \in V_{bf}, v = \{val(k), val(k+1), \ldots\}, \mathbf{card}(v) \leq p$
Ensure: $\mathbf{card}(O_p) \subseteq \mathbf{card}(O_f)$
  $IT = \emptyset$, $O_p = \emptyset$
  {Required Preference Test}
  if $\exists rname(i) \in O_f$ where $PreferenceArray(i,j,k) =$ Required **and**$(V_{bf}(j) = val(k)) == TRUE$ **then**
    $O_p = O_p \cup rname(i)$
  end if
  {Termination point 1}
  if $\mathbf{card}(O_p) = 1$ **and** $\exists PreferenceArray(i,\text{-},\text{-}) =$ {Prohibit} **then**
    $IT =$ {Constraint Failure} **and Terminate**
  else if $\mathbf{card}(O_p) > 1$ **then**
    $IT =$ {Constraint Failure} **and Terminate**
  else
    Terminate
  end if
  {Acceptable Preference Test}
  if $\exists rname(i) \in O_f$ where $PreferenceArray(i,j,k) =$ Acceptable **and**$(V_{bf}(j) = val(k)) == TRUE$ **then**
    $O_p = O_p \cup rname(i)$
  end if
  {Prohibited and Rejected Preference Test}
  if $\exists rname(i) \in O_p$ where $PreferenceArray(i,j,k) =$ {Prohibited} $\vee$ {Rejected} **and**$(V_{bf}(j) = val(k)) == TRUE$ **then**
    $O_p = O_p \setminus rname(i)$
  end if
  {Termination point 2}
  if $\mathbf{card}(O_p) < 1$ **then**
    $IT =$ {No Change} **and Terminate**
  else if $\mathbf{card}(O_p) = 1$ **then**
    Terminate
  end if
  {Remaining 4 tests and 3 terminations points can be found in [10].}
  return O_p, IT

[NBG21] Neogi, N., Bhattacharyya, S., Griessler, D., Kiran, H., and M. Carvalho, "Assuring Intelligent Systems: Contingency Management for UAS," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 6028-6038, Sept. 2021, doi: 10.1109/TITS.2021.3076399

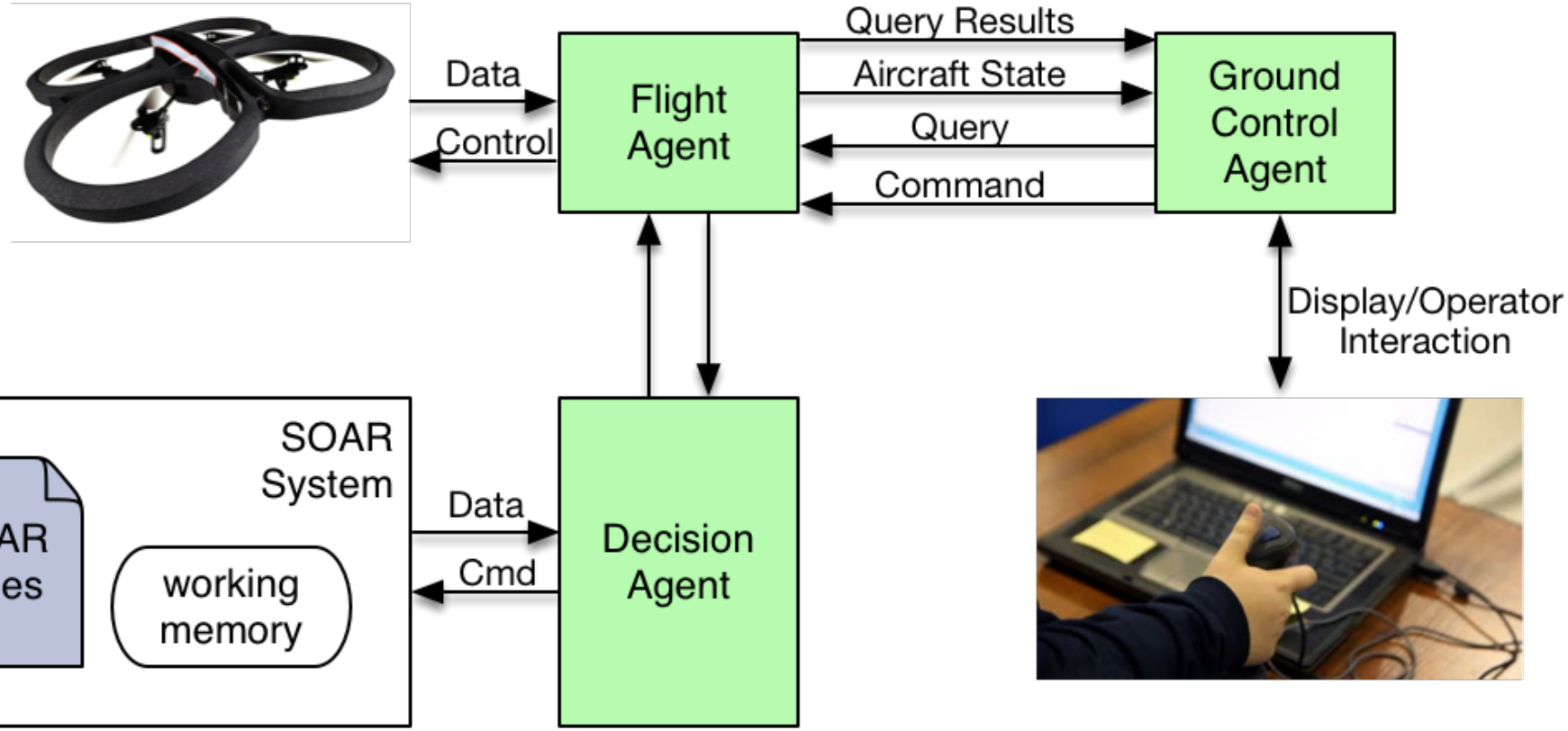
Translator Progress

- Translator between Soar and Uppaal
  - First iteration: flight simulator (simple Soar agent)
  - Second iteration: Missionaries and Cannibals (more complex Soar agent with learning but discrete problem space)
  - Third iteration: flight simulator (more complex Soar agent with learning simulating lost link to UAM)
- Verify and validate specific safety properties of a Soar agent using Uppaal

Outline

- Introduction and Motivation
- Urban Air Mobility Concept and Issues
- Verification and Validation Approach
- Case Study
- Conclusions and Future Work

Overview of Simulation Setup



Photos used with permission by D. Griessler, taken from Neogi, N., Bhattacharyya, S., Griessler, D., Kiran, H., and M. Carvalho, "Assuring Intelligent Systems: Contingency Management for UAS," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 6028-6038, Sept. 2021, doi: 10.1109/TITS.2021.3076399.

Contingency Management Approach: Lost Link

- Contingency behaviors implemented:

ID	Event	Environment (overflight population)	Contingency Behavior	Time to Execute
C1	Baseline		No Change	N/A
C2	Lost Link	Lightly populous→ Populous	Return to base	1 min
C3	Lost Link	Lightly populous	Return to base (or circle + re-establish)	8 min
C4	Lost Link	Unpopulated	Maintain Pre-Programmed Course, Return to mission altitude	1 min

Contingency Management Approach (II)

- Structure:



Normal:

Loop: get flightdata, update Soar input, catch Soar commands, send commands to XPlane

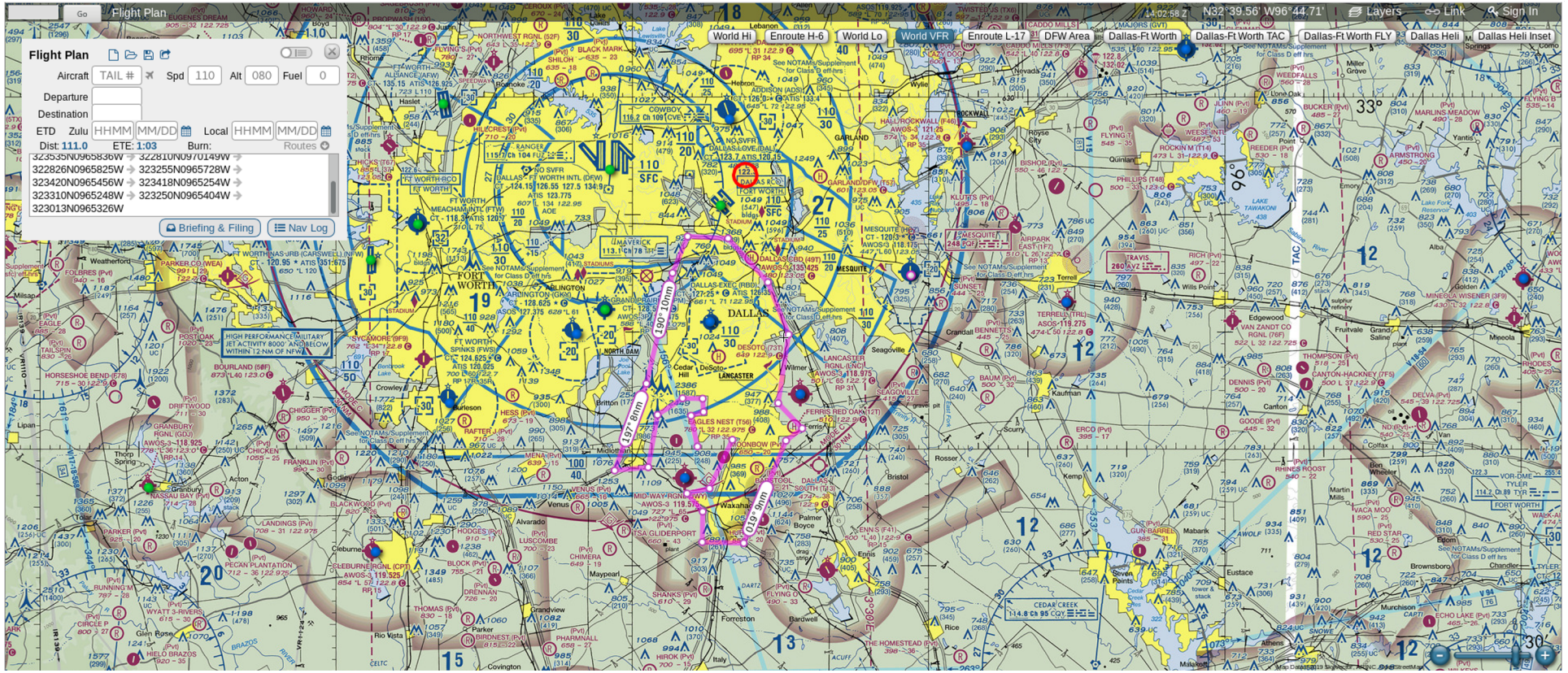
Find closest non-populated point that can be circled when needed

Contingency:

Loop: react to selected contingency scenerio, calculate equations needed for flight, use autopilot to guide airplane

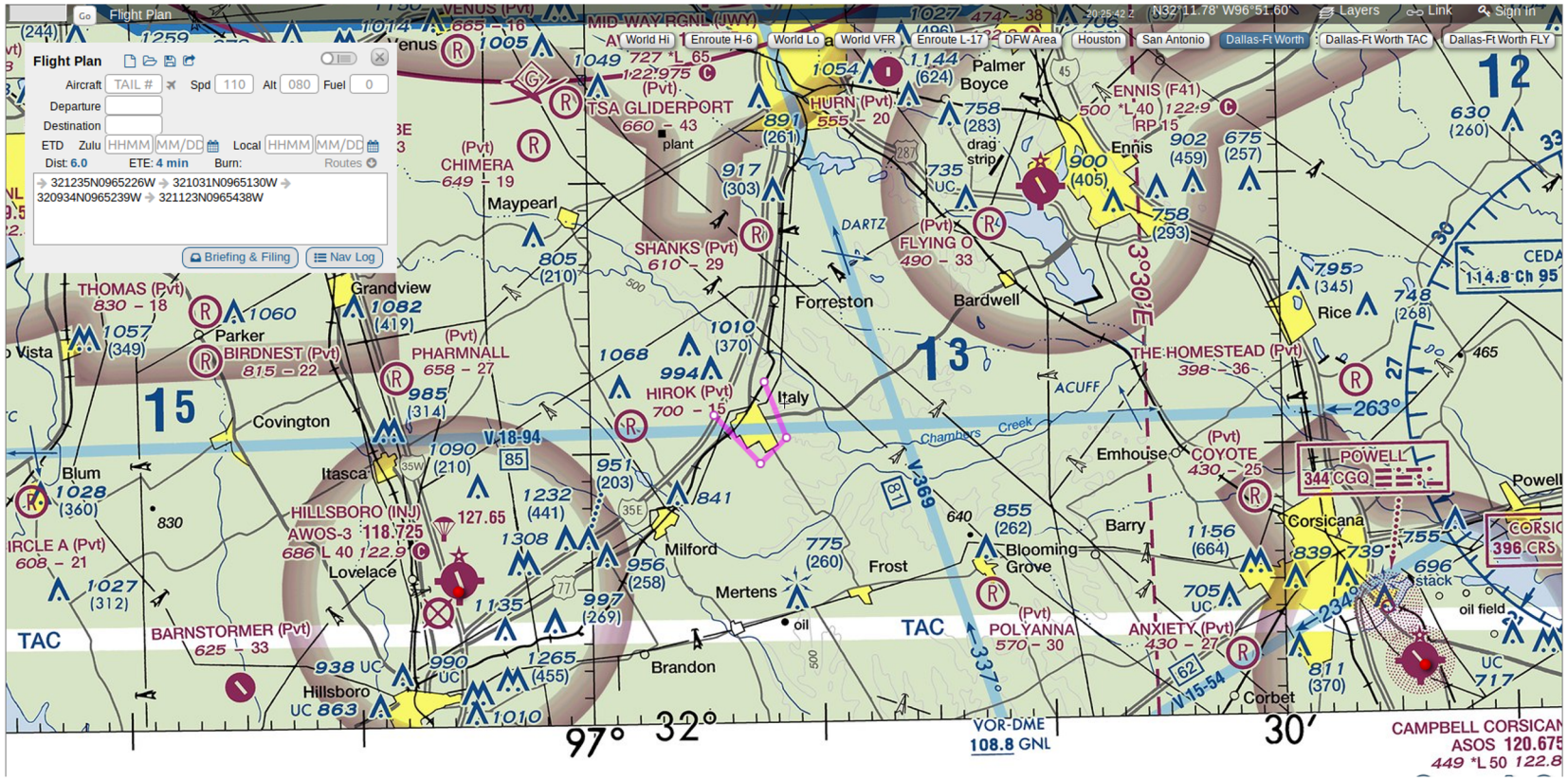
Handle contingency calculations as needed

Contingency Management Approach (III)



Source: Publicly Available FAA Sectional Charts, https://aeronav.faa.gov/visual/05-19-2022/PDFs/Dallas-Ft_Worth.pdf

Contingency Management Approach (IV)



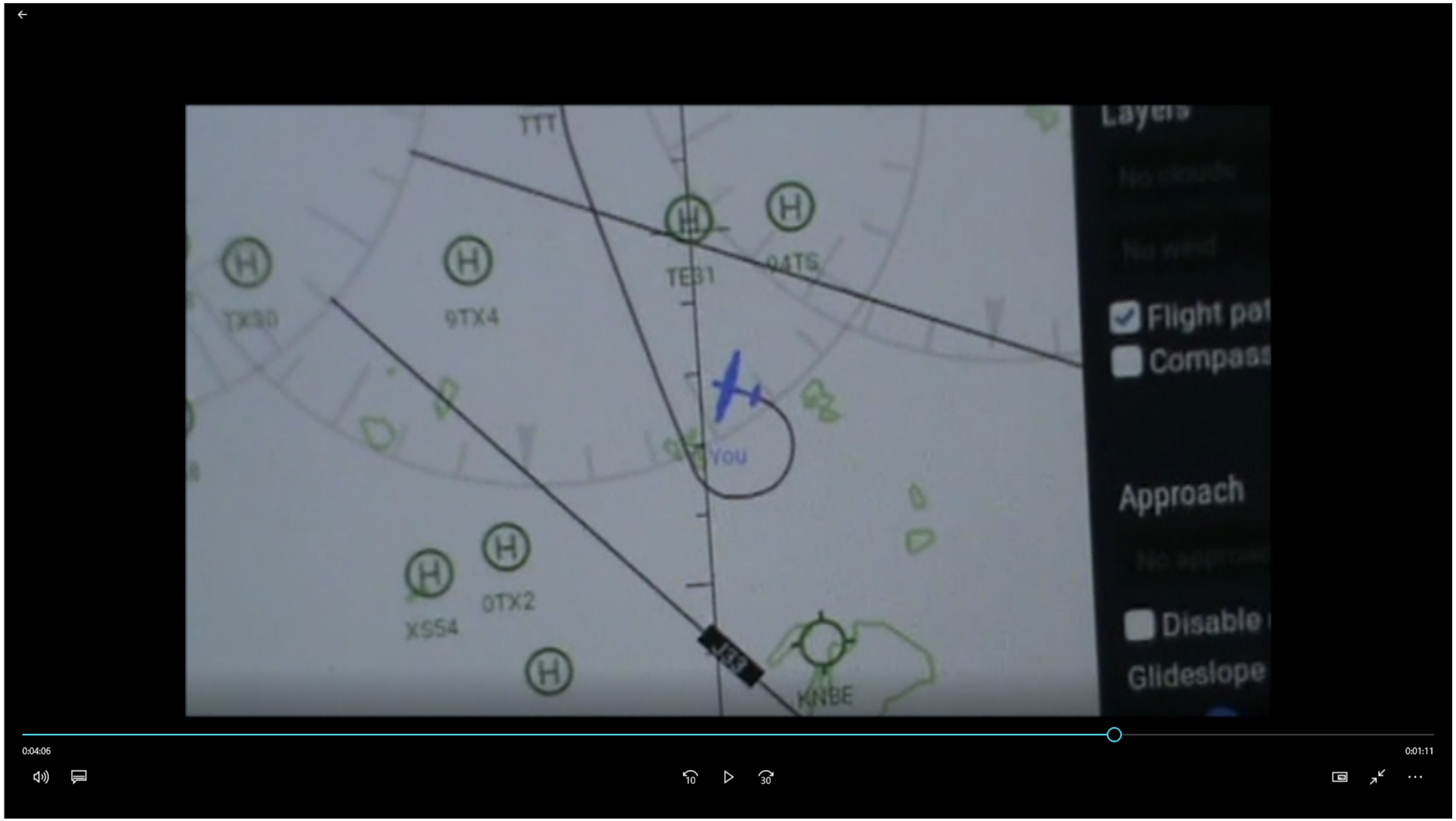
Source: Publicly Available FAA Sectional Charts, https://aeronav.faa.gov/visual/05-19-2022/PDFs/Dallas-Ft_Worth.pdf

Simulation (I)



Used with permission by D. Griessler, taken from Neogi, N., Bhattacharyya, S., Griessler, D., Kiran, H., and M. Carvalho, "Assuring Intelligent Systems: Contingency Management for UAS," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 6028-6038, Sept. 2021, doi: 10.1109/TITS.2021.3076399.

Simulation (II)



Used with permission by D. Griessler, taken from Neogi, N., Bhattacharyya, S., Griessler, D., Kiran, H., and M. Carvalho, "Assuring Intelligent Systems: Contingency Management for UAS," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 6028-6038, Sept. 2021, doi: 10.1109/TITS.2021.3076399.

Learning

- Learning is possible when flying over a lightly populated area due to initial conflict between C2 and C3
- Resolve: if both suggesting timers, select the one with less timer length
  - If not calculated to be in a populated area and no populated area is encountered, should have picked C3
  - If populated area encountered or flying over a populated area, then correct decision to pick C2

Learning (II)

- C2 runs and completes. Conflict: turn around or activate C3
- Resolve: turn around option is strictly worse than activating longer contingencies since no populated areas were encountered
  - If over the course of C3 a populated area is projected or encountered, then a new contingency can be learned and saved as a new rule

Learning New Rules

- Chunking – sort of. Time crunch

```
sp {drone*propose*operator*C6-Lost-Link-Start
    "In lightly populated area, should start a 2 minute timer and then turn around"
    (state <s> ^name droneFlight ^startContingency true -^contingencyComplete C6-Start ^io.input-link.flightdata <fd>)
    (<fd> ^takeOver true ^populated << lightly fully >>)
    -->
    (<s> ^operator <o> +)
    (<o> ^name C6-Start
         ^timerLength 2)
    (write (crlf) |PROPOSE C6!|)
}

sp {drone*propose*operator*C7-Lost-Link-Start
    "In lightly populated area, should start a 4 minute timer and then turn around"
    (state <s> ^name droneFlight ^startContingency true -^contingencyComplete C7-Start ^io.input-link.flightdata <fd>)
    (<fd> ^takeOver true ^populated << lightly fully >>)
    -->
    (<s> ^operator <o> +)
    (<o> ^name C7-Start
         ^timerLength 4)
    (write (crlf) |PROPOSE C7!|)
}
```

```
newRule =
        "\nsp {drone*propose*operator*C" + latestRuleNum + "-Lost-Link-Start\n" +
        "    \"In lightly populated area, should start a " + value + " minute timer and then turn around\"\n" +
        "    (state <s> ^name droneFlight ^startContingency true -^contingencyComplete " + newRuleName + " ^io.input-link.flightdata <fd>)\n" +
        "    (<fd> ^takeOver true ^populated << lightly fully >>)\n" +
        "    -->\n" +
        "    (<s> ^operator <o> +)\n" +
        "    (<o> ^name " + newRuleName + "\n" +
        "         ^timerLength " + value + ")\n" +
        "    (write (crlf) |PROPOSE C" + latestRuleNum + "!|)\n" +
        "}\n";
```

Learning from Saved Decisions [NBG21, SBC20, BEN18]

- Learning also possible through use of saved decisions
- As contingencies are activated over flight, save the decision for that position
- New decisions close to that point should use the closest saved decision instead of starting at C2 if there is one, otherwise go through learning process

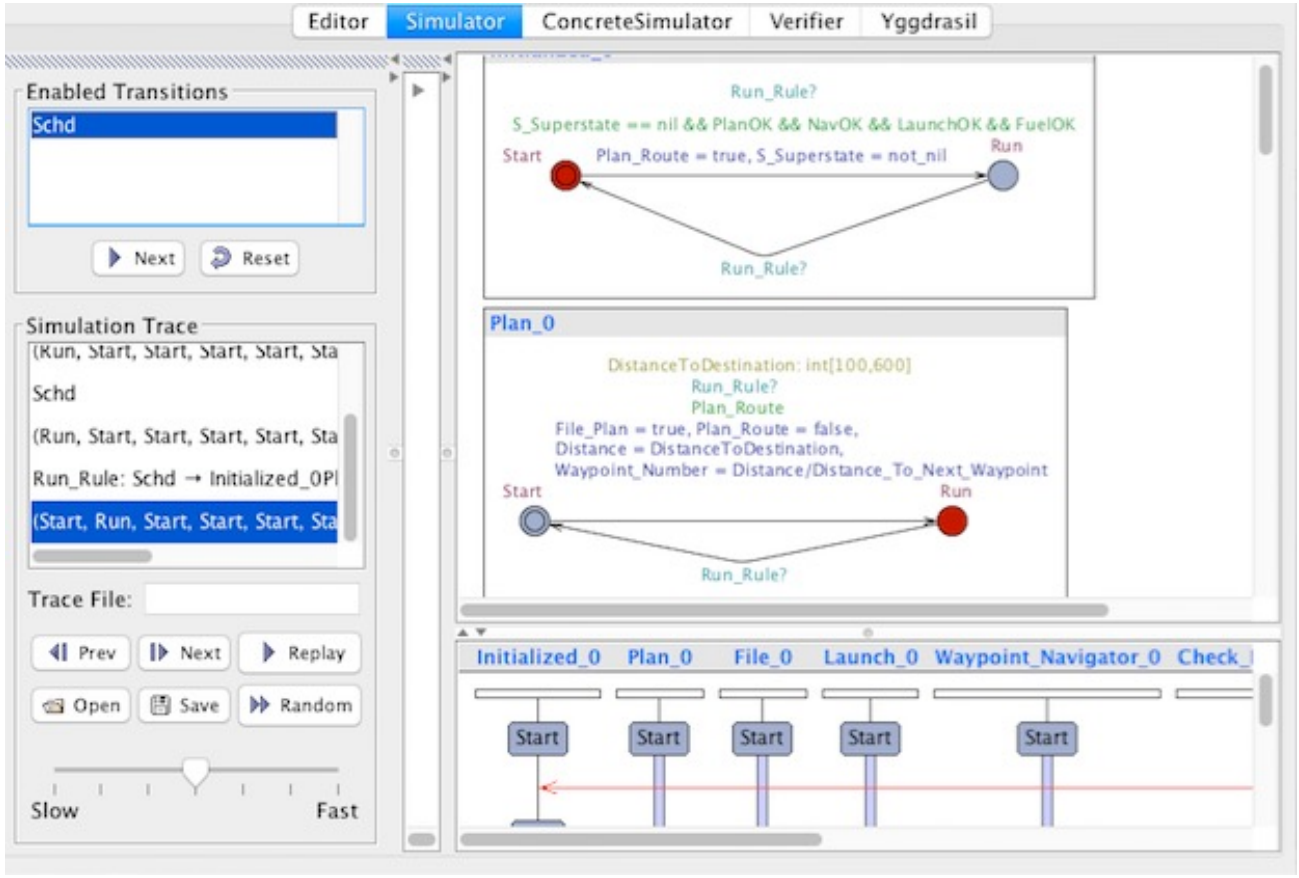
[NBG21] Neogi, N., Bhattacharyya, S., Griessler, D., Kiran, H., and M. Carvalho, "Assuring Intelligent Systems: Contingency Management for UAS," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 6028-6038, Sept. 2021, doi: 10.1109/TITS.2021.3076399.

[SBC20] Stafford, M., Bhattacharyya, S., Clark, M., Neogi N., and T. C. Eskridge, "Assurance for Integrating Advanced Algorithms in Autonomous Safety-Critical Systems," in IEEE Systems Journal, Dec 2020, doi: 10.1109/JSYST.2020.3023286.

[BEN18] Bhattacharyya, S., Eskridge, T., Neogi, N., Carvalho, M., and M. Stafford. “Formal Assurance for Cooperative Intelligent Autonomous Agents.” NFM 2018: 20-36.

Model Checking in UPPAAL for Learning Systems [BDL06]

- Modeling, validation and verification of real-time systems modeled as networks of timed automata, extended with data types (bounded integers, arrays, etc.)
  - Editor
  - Simulator
  - Verifier
- Verification of. 32 properties
  - Safety and Liveness
- Intermediate states checked through queries
- Simulation of counterexamples
- Visualization of timed automata



[BDL06] Behrmann, G., David, A., Larsen, K., Hakansson, J., Petterson, P., Yi, W., and Hendriks, M., (2006). UPPAAL 4.0. In Proceedings of the 3rd international conference on the Quantitative Evaluation of Systems (QEST '06). IEEE Computer Society, Washington, DC, USA, 125-126. DOI: https://doi.org/10.1109/QEST.2006.59

Outline

- Introduction and Motivation
- Urban Air Mobility Concept and Issues
- Verification and Validation Approach
- Case Study
- Conclusions and Future Work

Conclusions and Future Work

- Implemented translation mechanisms (in Soar to Uppaal translator) to support
  - Chunking with multi-valued attributes
  - Reinforcement Learning
- Simple safety and liveness properties related to multi-valued attributes and preferences verified
  - Enabled queries for intermediate states in learning process
- Performed case study on a ‘lost link’ protocol for UAM vehicles with degraded or denied communications
  - Learning applied to ‘wait and see’ protocols to activate appropriate timers
- Verification of complex safety and liveness properties in progress

References

[LBA17] Laird, J. E., Bates, C. C., Assanie, M., Derbinsky, N., & Xu, J. (2017). The soar user's manual version 9.6.0. Retrieved from https://soar.eecs.umich.edu/downloads/SoarManual.pdf

[LLR06] Lehman, J., Laird, J., & Rosenbloom, P. (2006). A gentle introduction to soar, an architecture for human cognition: 2006 update., 15. Retrieved from http://web.eecs.umich.edu/~soar/sitemaker/docs/misc/GentleIntroduction-2006.pdf

[AD94] Alur, R. and Dill, D. (1994). A theory of timed automata. Theor. Comput. Sci. 126, 2 (April 1994), 183-235. DOI=10.1016/0304-3975(94)90010-8 http://dx.doi.org/10.1016/0304-3975(94)90010-8

[BDL06] Behrmann, G., David, A., Larsen, K., Hakansson, J., Petterson, P., Yi, W., and Hendriks, M., (2006). UPPAAL 4.0. In Proceedings of the 3rd international conference on the Quantitative Evaluation of Systems (QEST '06). IEEE Computer Society, Washington, DC, USA, 125-126. DOI: https://doi.org/10.1109/QEST.2006.59

[BWN21] Bharadwaj S., Wongpiromsarn T., Neogi N., Muffoletto J., Topcu U. (2021) Minimum-Violation Traffic Management for Urban Air Mobility. In: Dutle A., Moscato M.M., Titolo L., Muñoz C.A., Perez I. (eds) NASA Formal Methods. NFM 2021. Lecture Notes in Computer Science, vol 12673. Springer, Cham. https://doi.org/10.1007/978-3-030-76384-8_3

[NBG21] Neogi, N., Bhattacharyya, S., Griessler, D., Kiran, H., and M. Carvalho, "Assuring Intelligent Systems: Contingency Management for UAS," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 6028-6038, Sept. 2021, doi: 10.1109/TITS.2021.3076399.

[BCN21] Bharadwaj, S., Carr, S., Neogi and U. Topcu, "Decentralized Control Synthesis for Air Traffic Management in Urban Air Mobility", in IEEE Transactions on Control of Network Systems, vol. 8, no. 2, pp. 598-608, June 2021, doi: 10.1109/TCNS.2021.3059847.

[SBC20] Stafford, M., Bhattacharyya, S., Clark, M., Neogi N., and T. C. Eskridge, "Assurance for Integrating Advanced Algorithms in Autonomous Safety-Critical Systems," in IEEE Systems Journal, Dec 2020, doi: 10.1109/JSYST.2020.3023286.

[BEN18] Bhattacharyya, S., Eskridge, T., Neogi, N., Carvalho, M., and M. Stafford. "Formal Assurance for Cooperative Intelligent Autonomous Agents." NFM 2018: 20-36.

NASA Opportunities

- NASA Internships, Fellowships and Scholarships (NIFS) Program
- You must be/have
  - U.S. Citizenship
  - Cumulative 3.0 GPA (on a 4.0 scale)
  - Full-time students (high school through graduate)
  - Enrollment in a degree granting institution
  - 16 years of age at the time of application (no exceptions)
- There are three sessions:
  - Fall: Late August/early September, mid-December, 16 weeks, July 1 application deadline
  - Spring: Mid-January, early May, 16 weeks, November 1 application deadline
  - Summer: Late May/early June, August, 10 weeks, April 1 application deadline

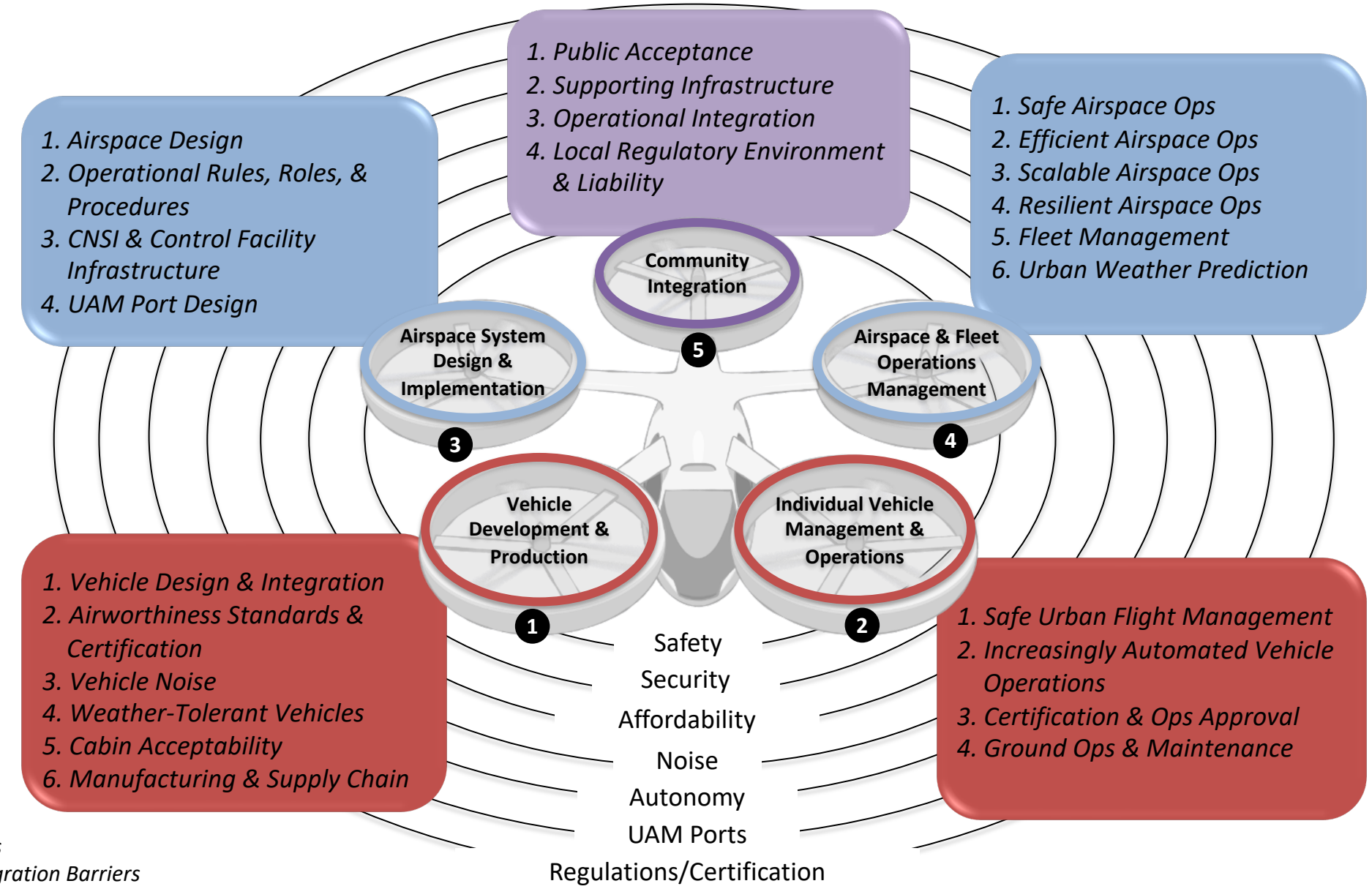
https://intern.nasa.gov

Thanks to Dr. Siddhartha Bhattacharyya, Daniel Greissler and Milton Stafford at the Florida Institute of Technology

QUESTIONS?

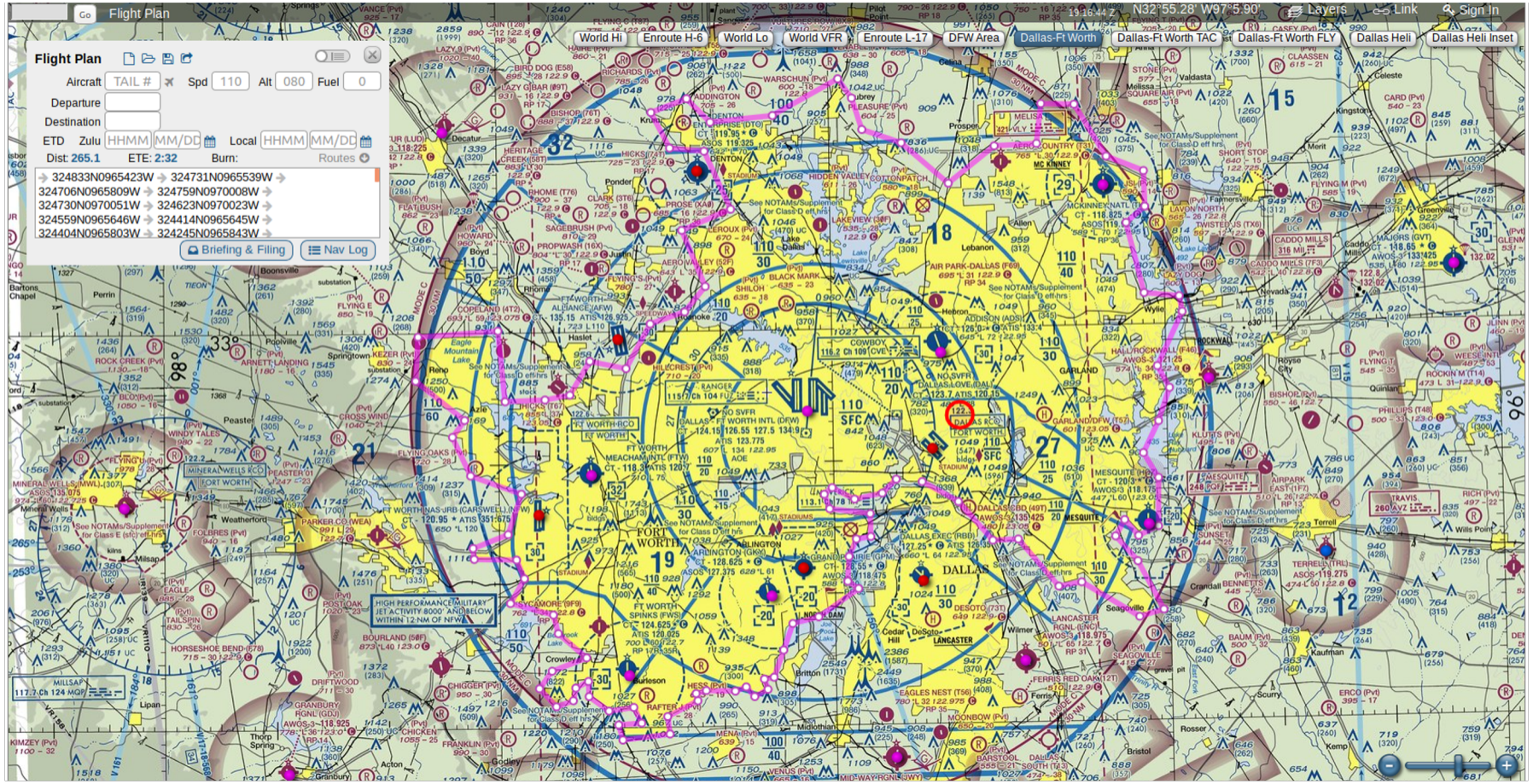
Natasha.A.Neogi@nasa.gov

UAM Framework and Barriers



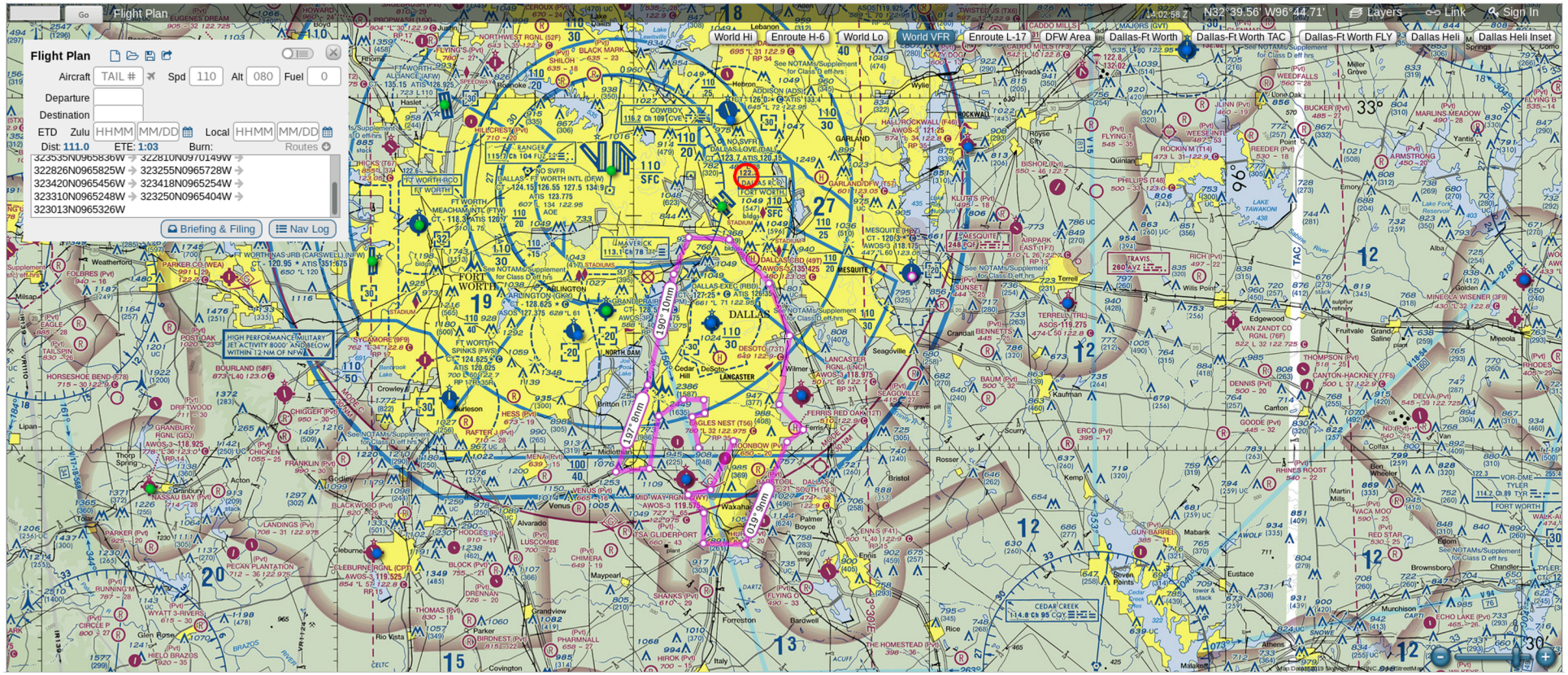
Vehicle Barriers
Airspace Barriers
Community Integration Barriers
\# *Pillar number*

Contingency Management Approach (III)



Source: Publicly Available FAA Sectional Charts, https://aeronav.faa.gov/visual/05-19-2022/PDFs/Dallas-Ft_Worth.pdf

Contingency Management Approach (IV)



Source: Publicly Available FAA Sectional Charts, https://aeronav.faa.gov/visual/05-19-2022/PDFs/Dallas-Ft_Worth.pdf