Ames Autonomous Systems Assurance

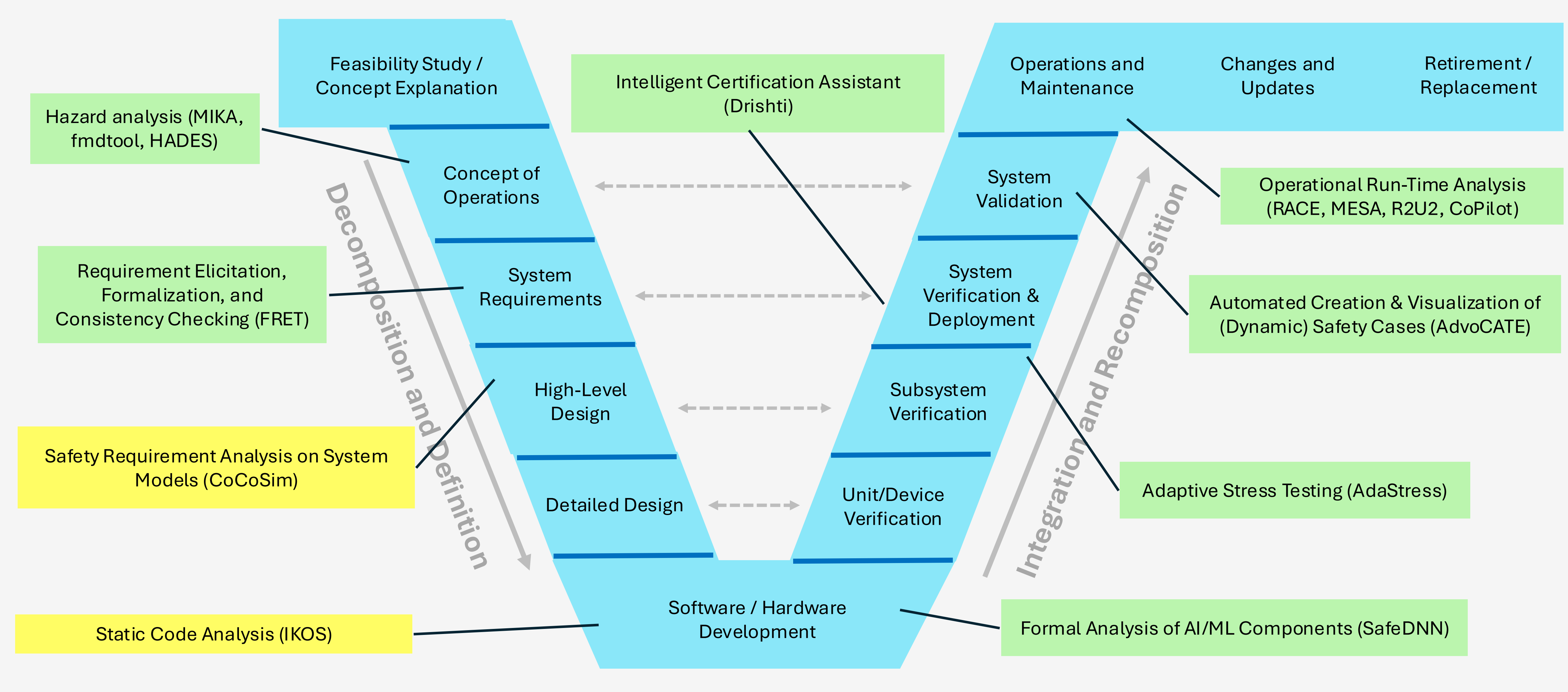
TC4 Closeout Workshop

Dr. Guillaume Brat
Robust Software Engineering
Intelligent Systems Division
NASA Ames Research Center

Global Accomplishments

- Created (and updated) 10+ assurance tools targeting autonomous systems, especially those enabled by machine learning or AI.
    - Every stage of the V development process has been targeted
- Worked with industry and academia to get requirements and real-world data, and, to validate the tools
    - Industrial collaborators include Rockwell Collins., General Electric, Boeing, Joby (ex-Xwing), Reliable Robotics, Wisk, Nuro, and Google Loons, and more.
- Worked with government agencies to address current regulations
    - Collaboration with Dr. Trung Pham (FAA) on Dr. Huafeng Yu (DoT) on guidelines drafted by FAA or by SAE G34 committee
    - Worked with DoD on draft for their T&E guidelines for autonomous systems

V diagram for ARC



Feasibility Study / Concept Explanation

Concept of Operations

System Requirements

High-Level Design

Detailed Design

Software / Hardware Development

Unit/Device Verification

Subsystem Verification

System Verification & Deployment

System Validation

Operations and Maintenance

Changes and Updates

Retirement / Replacement

Decomposition and Definition

Integration and Recomposition

Hazard analysis (MIKA, fmdtool, HADES)

Requirement Elicitation, Formalization, and Consistency Checking (FRET)

Safety Requirement Analysis on System Models (CoCoSim)

Static Code Analysis (IKOS)

Intelligent Certification Assistant (Drishti)

Operational Run-Time Analysis (RACE, MESA, R2U2, CoPilot)

Automated Creation & Visualization of (Dynamic) Safety Cases (AdvoCATE)

Adaptive Stress Testing (AdaStress)

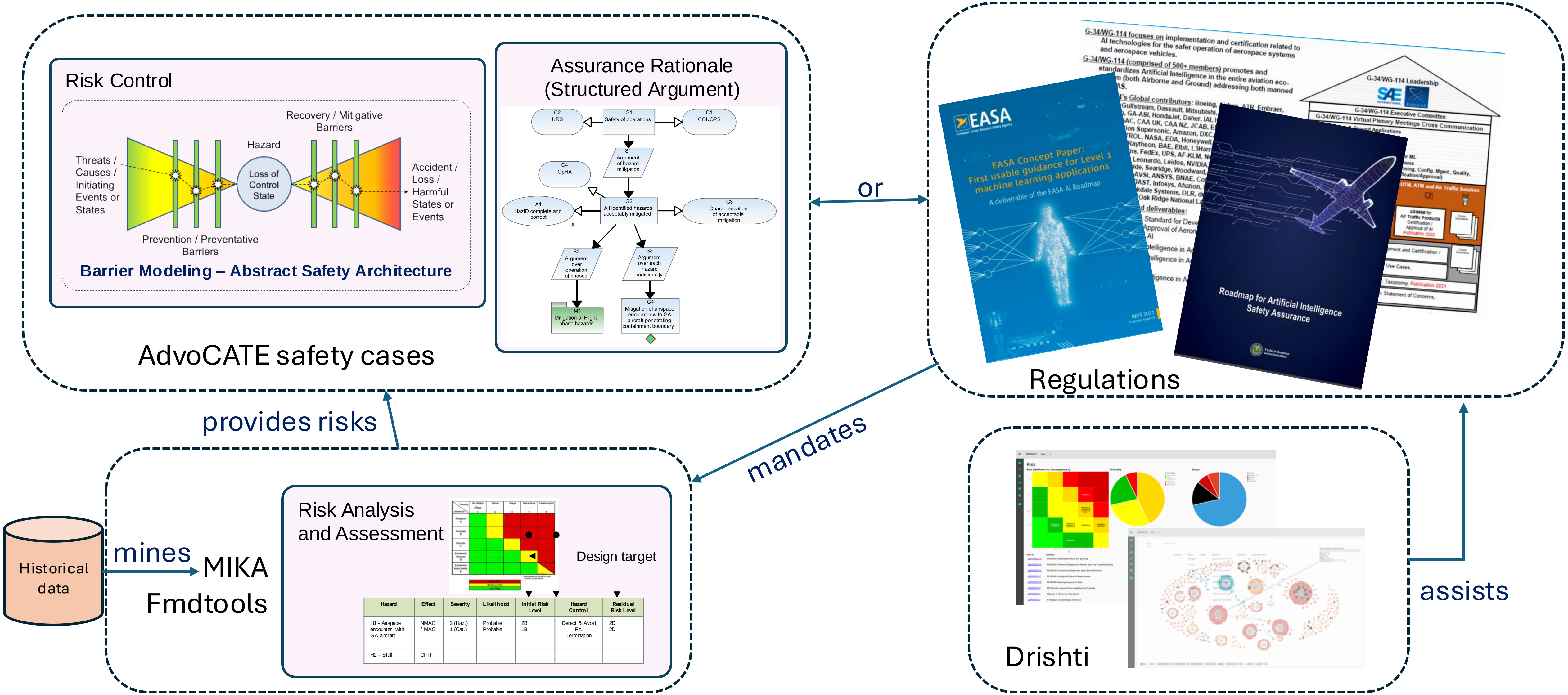
Formal Analysis of AI/ML Components (SafeDNN)

Hazard Analysis, Process & Certification

- SAE G-34 Participation
  - Specific inputs on data-centric Operational Design Domain
  - Active participation in committee meetings
  - Support for our FAA colleagues
- AdvoCATE dynamic dashboard
  - Continued development of tool created in TC3 for authoring safety cases
  - Extension to use safety cases in operations (dynamic safety cases)
- Drishti
  - Intelligent search to create material for certification reviews
- Hazard analysis
  - Mining historical data for risks
  - Simulation tools to explore complex failures to identify more risks

Project	Approach	Researchers
SAE G-34	Active meeting participation ODD research	Ganesh Pai
AdvoCATE	Assurance cases Dynamic safety cases	Ewen Denney Ganesh Pai Irfan Sjlivo
Drishti	Intelligent certification assistant	Nija Shi
MIKA fmdtool	Mining historical data Simulation for resilience	Hannah Walsh Sequioa Andrade Daniel Hulse Lukman Irshad

Example use


Risk Control
Threats / Causes / Initiating Events or States
Prevention / Preventative Barriers
Hazard
Loss of Control State
Recovery / Mitigative Barriers
Accident / Loss / Harmful States or Events
Barrier Modeling – Abstract Safety Architecture
Assurance Rationale (Structured Argument)
C2 URS
G1 Safety of operations
C1 CONOPS
S1 Argument of hazard mitigation
C4 OpHA
A1 HazID complete and correct
A
G2 All identified hazards acceptably mitigated
C3 Characterization of acceptable mitigation
S2 Argument over operational phases
S3 Argument over each hazard individually
M1 Mitigation of Flight-phase hazards
G4 Mitigation of airspace encounter with GA aircraft penetrating containment boundary
AdvoCATE safety cases
or
EASA Concept Paper: First usable guidance for Level 1 machine learning applications
A deliverable of the EASA AI Roadmap
April 2021
Roadmap for Artificial Intelligence Safety Assurance
Federal Aviation Administration
G-34/WG-114 Leadership
G-34/WG-114 Executive Committee
G-34/WG-114 Virtual Plenary Meetings Cross Communication
Regulations
provides risks
mandates
assists
Historical data
mines
MIKA
Fmdtools
Risk Analysis and Assessment
Design target
Drishti

Hazard	Effect	Severity	Likelihood	Initial Risk Level	Hazard Control	Residual Risk Level
H1 - Airspace encounter with GA aircraft	NMAC / MAC	2 (Haz.) 1 (Cat.)	Probable Probable	2B 1B	Detect & Avoid Flt. Termination …	2D 2D
H2 – Stall	CFIT					

Requirements & Runtime monitoring

- FRET enables developers to author and formalize requirements
    - Yields clear, unambiguous, formalized requirements
    - Supports formal analysis of requirements (e.g., consistency checking)
    - Supports automatic generation of test scenarios
    - Now supports authoring of probabilistic requirements for autonomy
- Enables formal analysis further in the lifecycle
    - Connects with CoCoSim (TC3) for Simulink model analysis
    - Connects with CoPilot and R2U2 to enable runtime monitoring based on formal requirements
- Runtime monitoring can be used to enable requirement based-testing and monitoring of safety conditions during operations

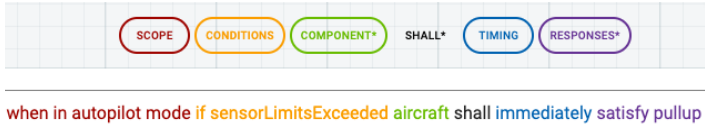
Project	Approach	Researchers
FRET	Requirement formalization Consistency checking Test scenario generation	Anastasia Mavridou Andreas Katis
R2U2	Runtime monitoring Security checking	Johann Schumann
OGMA	Runtime monitoring	Ivan Perez

Example use

The mission of the Formal Requirement Elicitation Tool (FRET) is to:

- Provide *an intuitive platform* for capturing *precise* requirements.
- Enable *early V&V* during requirements elicitation and authoring phases.
- Serve as a *enabler* to a variety of external analysis tools.

Input: Intuitive restricted English

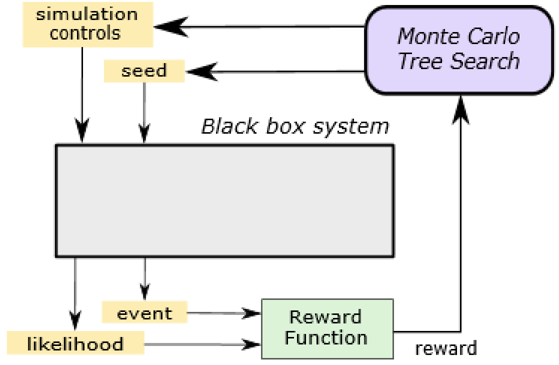


when in autopilot mode if sensorLimitsExceeded aircraft shall immediately satisfy pullup

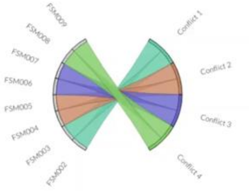
Output: Complex formal logics

```
((G ((! ((! autopilot) & (X autopilot))) | (X
(((autopilot & (X (! autopilot))) V (((!
sensorLimitsExceeded) & ((X sensorLimitsExceeded) & (!
(autopilot & (X (! autopilot)))))) -> ((X pullup) & (!
(autopilot & (X (! autopilot))))))) &
(sensorLimitsExceeded -> pullup))))) & (autopilot ->
(((autopilot & (X (! autopilot))) V (((!
sensorLimitsExceeded) & ((X sensorLimitsExceeded) & (!
(autopilot & (X (! autopilot)))))) -> ((X pullup) & (!
(autopilot & (X (! autopilot))))))) &
(sensorLimitsExceeded -> pullup))))
```

Adaptive stress testing

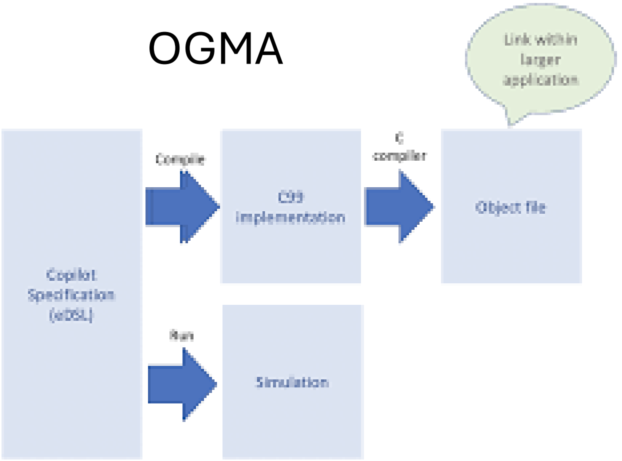


Consistency checking

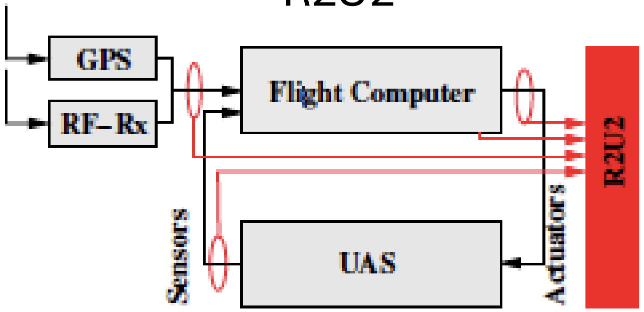


Formal runtime monitors

OGMA

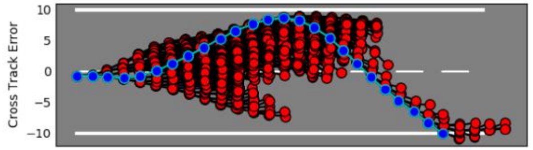


R2U2

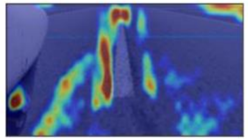


Autonomous Systems V&V

- SafeDNN: Formal analysis of ML components
  - Formal analysis of DNN models to infer properties which could be used for understanding, verifying, debugging and testing
  - Using confusion matrices as abstraction for ML-components and support system-level compositional verification
- Adaptive stress testing
  - AdaStress is a software package for an accelerated simulation-based stress testing method for finding the most likely path to a failure event
  - Use Reinforcement Learning techniques to drive testing towards rare failure events
  - Integrated by GE in their testing tool suite
- SysAI
  - Provides advanced capabilities that support understanding the system behavior in nominal and off-nominal situations (e.g., safe operational regions)



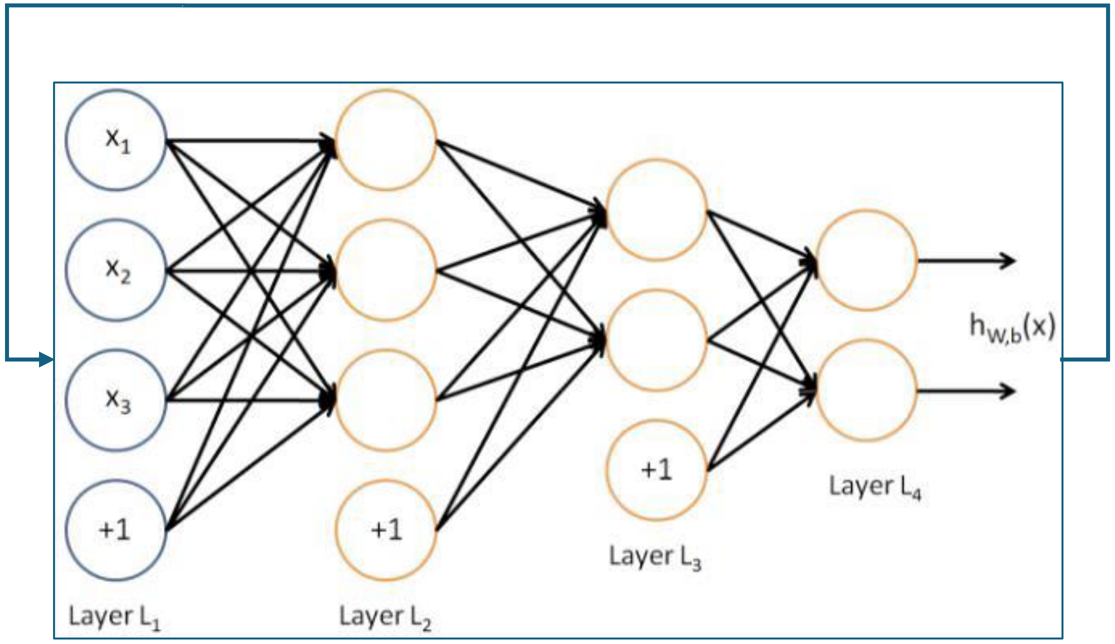
Project	V&V approach	Researchers
SafeDNN	Formal methods Rule inference Neural network analysis	Divya Gopinath Corina Păsăreanu
SYSAI	Bayesian methods Boundary analysis	Yuning He
AdaStress	Statistical methods Reinforcement learning Black-box analysis	Adrian Agogino Rory Lipkis

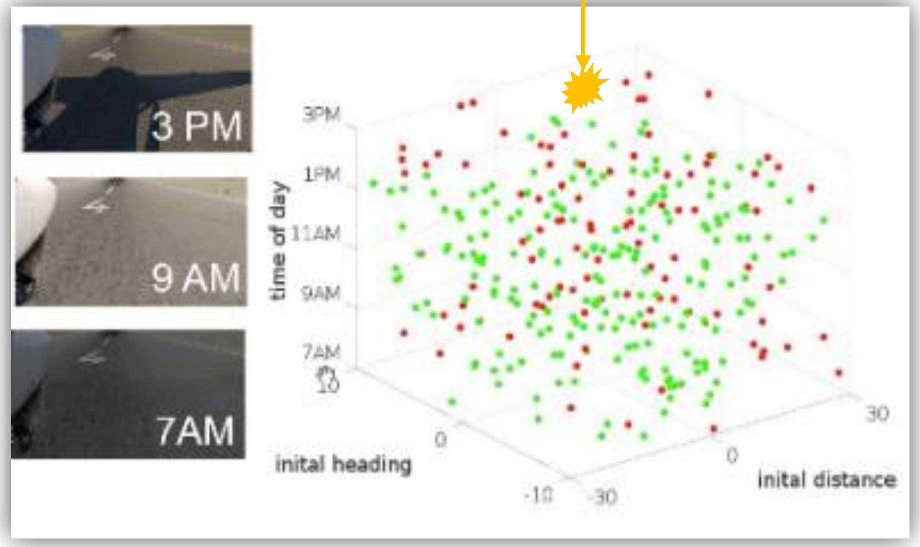


Example use

SysAI: Statistical learning framework to identify safe regions of operations

AdaStress: Reinforcement learning framework to identify rare event





Safe inputs

SafeDNN formal backward analysis

Desired output

RSE Software Engineering Tools

Tools	Description	Availability	Technical POC	POC Email
MIKA	Hazard analysis based on history	Open Source: https://github.com/nasa/mika	Hannah Walsh	hannah.walsh@nasa.gov
FmdTool	Resilience Analysis	Open Source: https://github.com/nasa/fmdtools	Daniel Hulse	daniel.e.hulse@nasa.gov
HADES	Hazard Analysis for Complex Systems	Not ready	Lukman Irshad	lukman.irshad@nasa.gov
CoPilot/OGMA	Runtime Verification	Open Source: https://copilot-language.github.io/ https://github.com/nasa/ogma	Ivan Perez	ivan.perezdominguez@nasa.gov
Drishti	Intelligent Assistant for Certification	Contact us	Nija Shi	nija.shi@nasa.gov
AdaStress	Adaptive stress testing	Open Source: https://github.com/NASA-SW-VnV/AdaStress.jl	Rory Lipkis	rory.lipkis@nasa.gov
SafeDNN	Formal analysis of Neural Networks	Open Source in process	Corina Pasareanu	corina.s.pasareanu@nasa.gov
R2U2	Vehicle-level run-time analysis	Usage Agreement	Johann Schumann	johann.m.schumann@nasa.gov

RSE Software Engineering Tools

Tools	Description	Availability	Technical POC	POC Email
FRET	Requirement elicitation and analysis	Open Source: https://github.com/NASA-SW-VnV/fret	Anastasia Mavridou	anastasia.mavridou@nasa.gov
CoCoSim	Simulink model analyzer	Open Source: https://github.com/NASA-SW-VnV/CoCoSim	Andreas Katis	andreas.katis@nasa.gov
IKOS	Static code analysis for C/C++	Open Source: https://github.com/NASA-SW-VnV/ikos	Ivan Perez	ivan.perezdominguez@nasa.gov
AdvoCATE	Assurance case automation toolset	Open Source: contact POC	Ewen Denney	ewen.w.denney@nasa.gov
MARGInS	ML/statistical libraries for system testing	Usage Agreement	Carlos Paradis	carlos.v.paradis@nasa.gov
SysAI	ML/statistical libraries for system testing	Contact us	Yuning He	yuning.he@nasa.gov
RACE-ODIN	Runtime for Airspace Concept Evaluation	Open Source: https://nasarace.github.io/race-odin/	Peter Mehlitz	peter.c.mehlitz@nasa.gov
MESA	Run-time analysis of live data streams	Open Source: https://github.com/NASA-SW-VnV/mesa	Not maintained	N/A

Impact

Tools	Industry	AOSP	NASA	International	Others
AdaStress	GE, General Atomics		STMD	Zaebuz (Norway), Norwegian University of Science and Tech (NTNU), CERN	FAA, DARPA
AdvoCATE	Many (Boeing, …)	ACERO	SMD, IV&V	Boeing Australia, Universities in UK	FAA, DARPA
Drishti	Interest from GE		STMD		
Fmdtool **MIKA**	Boeing			GRC (ISAT)	AFRL, DoT
FRET	Lockheed, RTX Tech center, Galois, GE	UTM	ARC, JPL, LaRC	CERN, Universities in UK/Spain/Portugal, Bosch, JAXA, Collins Ireland	FAA, NRC, NREL, DARPA (thru RTX), Stanford
ODIN	Delphire		ARMD, ESMD	North Holland region	USFS, Santa Clara county
OGMA				Universities in Spain	
R2U2	Boeing		JSC, ARC	DLR (Germany)	University of Iowa
SafeDNN	Boeing, VMWare		STMD	Universities in UK/Canada	DoT, FAA, SRI, universities (Stanford, Berkeley, CMU, Virginia, UT Austin)
SysAI	Boeing				FAA, DoT

Conclusions

- Created (and updated) 10+ assurance tools targeting autonomous systems, especially those enabled by machine learning or AI.
  - Every stage of the V development process has been targeted
- Worked with industry and academia to get requirements and real-world data, and, to validate the tools
  - Industrial collaborators include Rockwell Collins., General Electric, Boeing, Joby (ex-Xwing), Reliable Robotics, Wisk, Nuro, and Google Loons, and more.
- Worked with government agencies to address current regulations
  - Collaboration with Dr. Trung Pham (FAA) on Dr. Huafeng Yu (DoT) on guidelines drafted by FAA or by SAE G34 committee
  - Gave inputs to DoD on draft for their T&E guidelines for autonomous systems
- Working with industry and government agencies to identify future research.