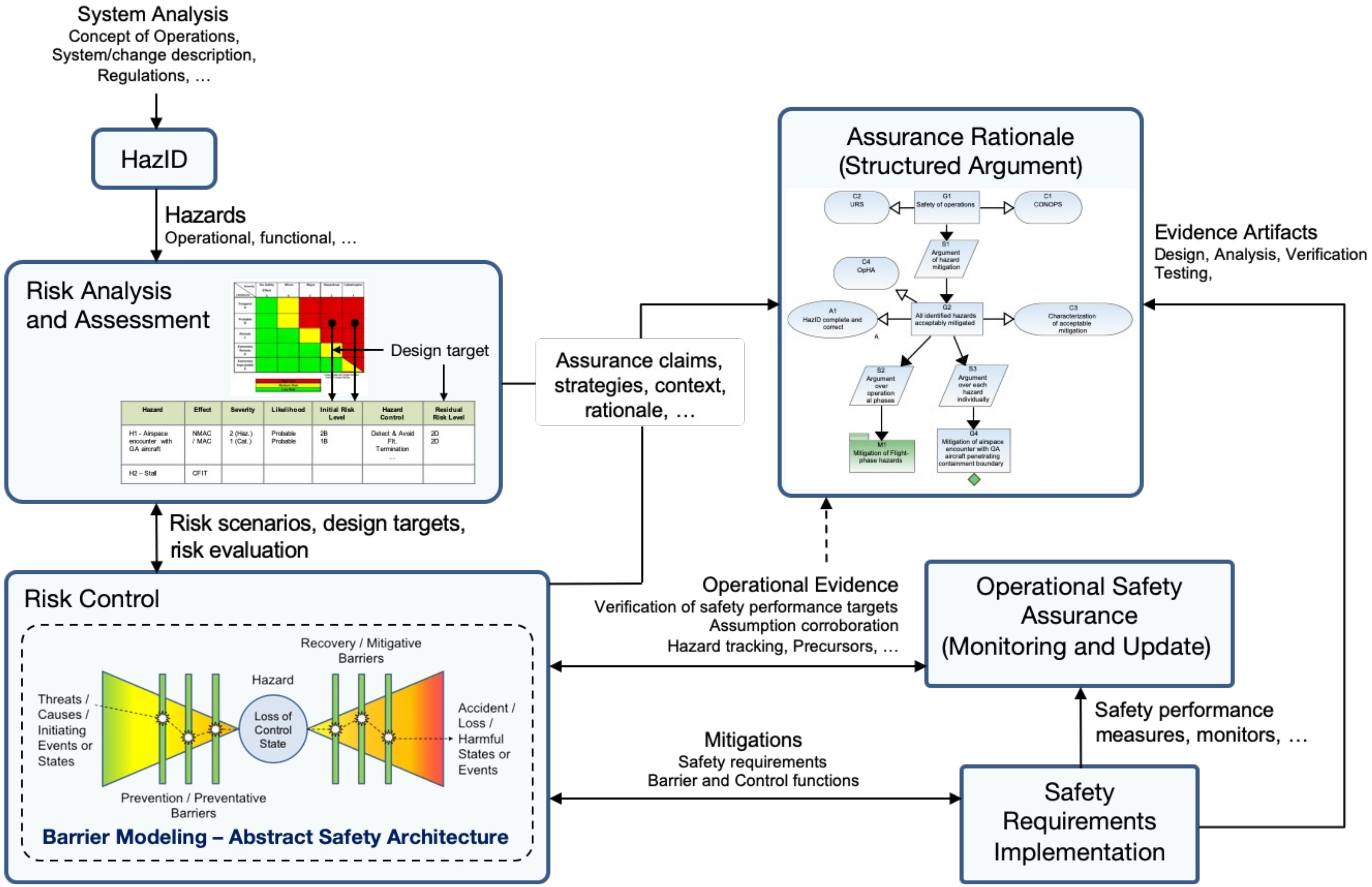
Dynamic Assurance Cases: Closing the Loop between Design and Operational Assurance

Ewen Denney, Jonathan Menzies, Ganesh Pai

KBR / NASA Ames Research Center

Software Certification Consortium, Meeting 21
May 11, 2023. Annapolis, MD

Assurance Methodology



- Focus of earlier work
  - Development of
    - Safety architectures
    - Arguments
    - Runtime risk assessment
- Current work
  - Safety performance indicators
  - Assurance case update

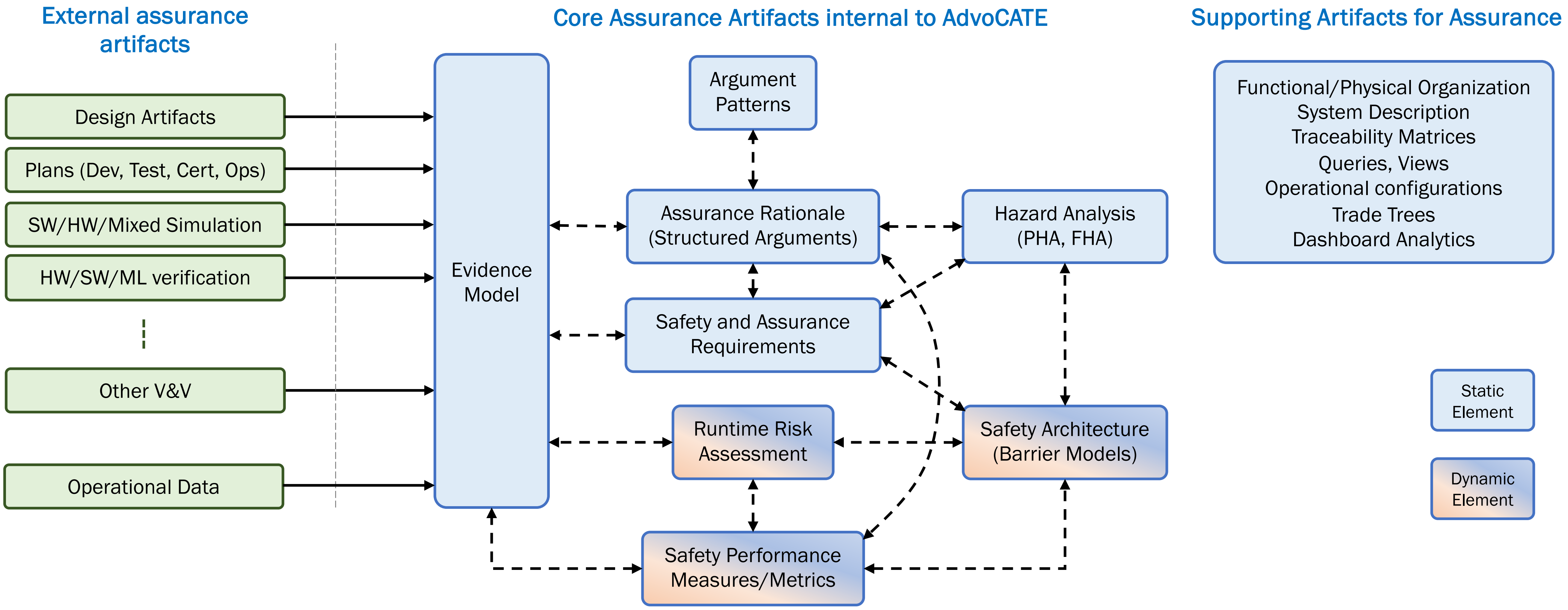
Static vs Dynamic Assurance

- Traditional assurance cases are static
  - Design time assurance
  - Establish a level of confidence to approve a system for service
  - Arguments should not be frequently updated
    - Argument update after operation → Argument was invalid to begin
- Dynamic assurance
  - Confirmation in operation that safety / assurance baseline is maintained (or improved)
  - Need a computable notion of assurance
    - Safety Architectures
      - (Compositions of) event-chain / barrier models connected to safety targets a.k.a. target level of safety (TLOS)

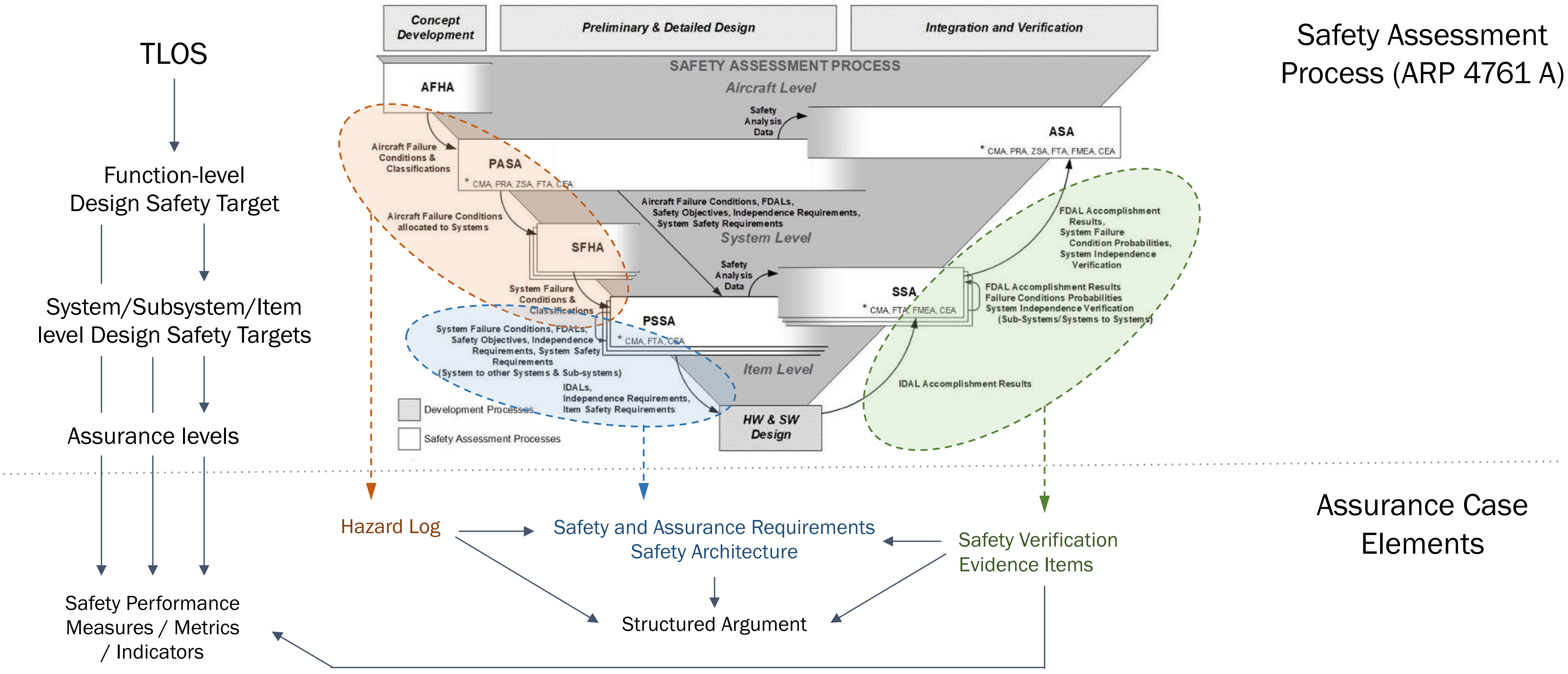
- Dynamic Assurance
  - Safety Architectures
    - Model for risk assessment
      - The goal is insight, not numbers
    - Linking design to operations
  - Runtime risk assessment models
  - Safety performance metrics and indicators
    - Relate to event and barrier risk levels
    - Relate to other assurance artifacts
  - Need to define what is updated, when, and how frequently
    - During missions (inner loop)
    - Between missions (outer loop)

Dynamic Assurance Case Concept

Multi-viewpoint, multi-artifact, model-based

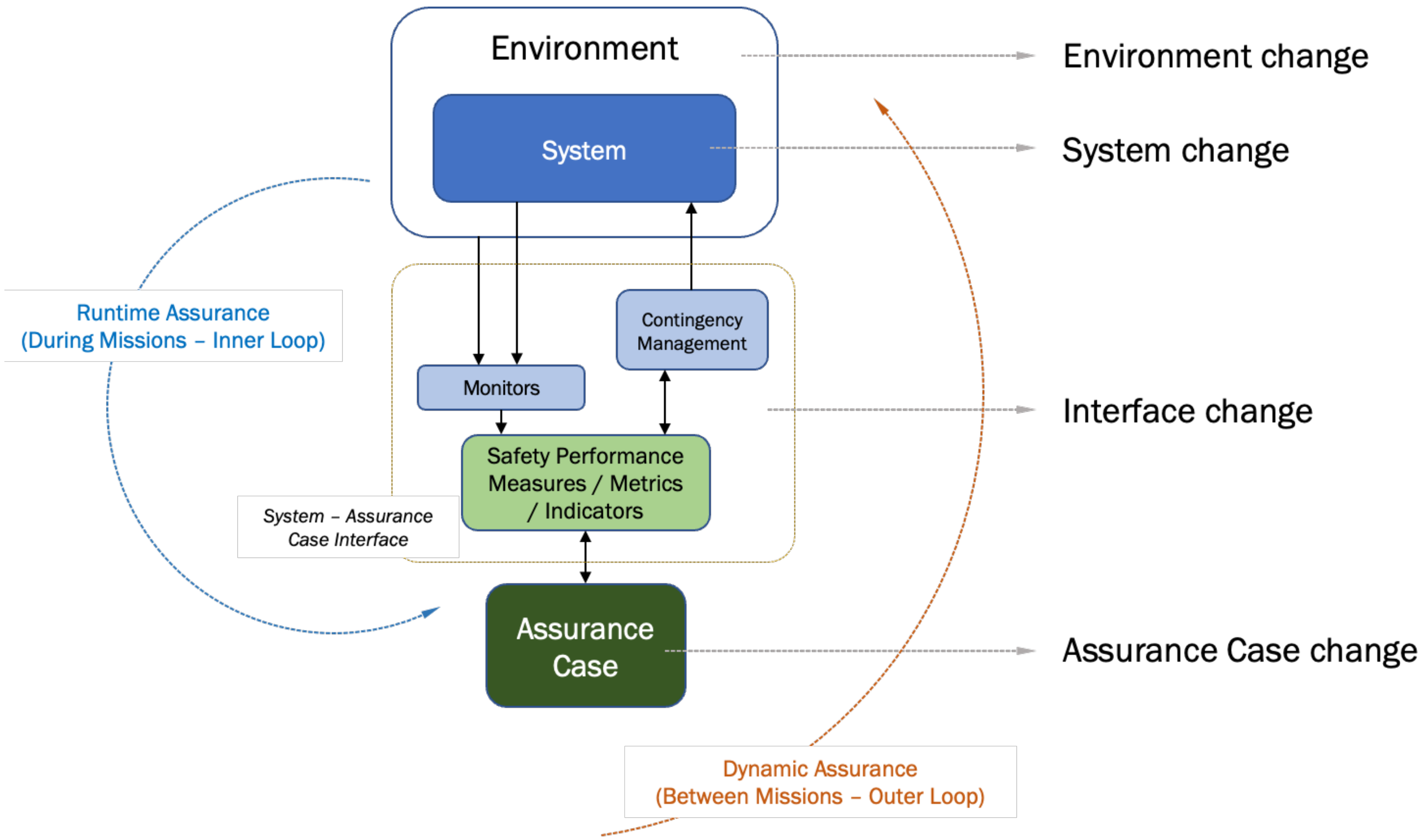


Linking Assurance Cases to Safety Processes

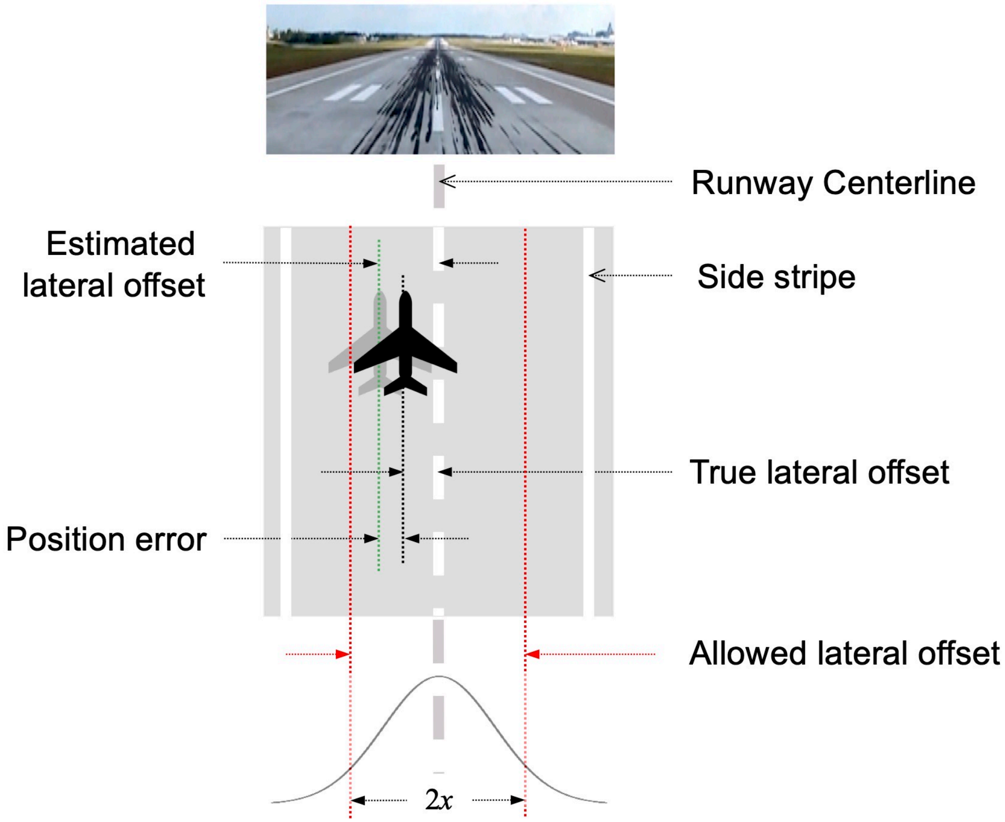


Dynamic Assurance and Update Concept

- System – Assurance Case Interface
  - Measure and modulate safety performance in operation
  - Ensure system stays within approved risk baseline
  - Operational verification of established safety performance targets
  - Triggers for runtime/dynamic assurance and corresponding updates



Motivating Example – Autonomous Taxiing

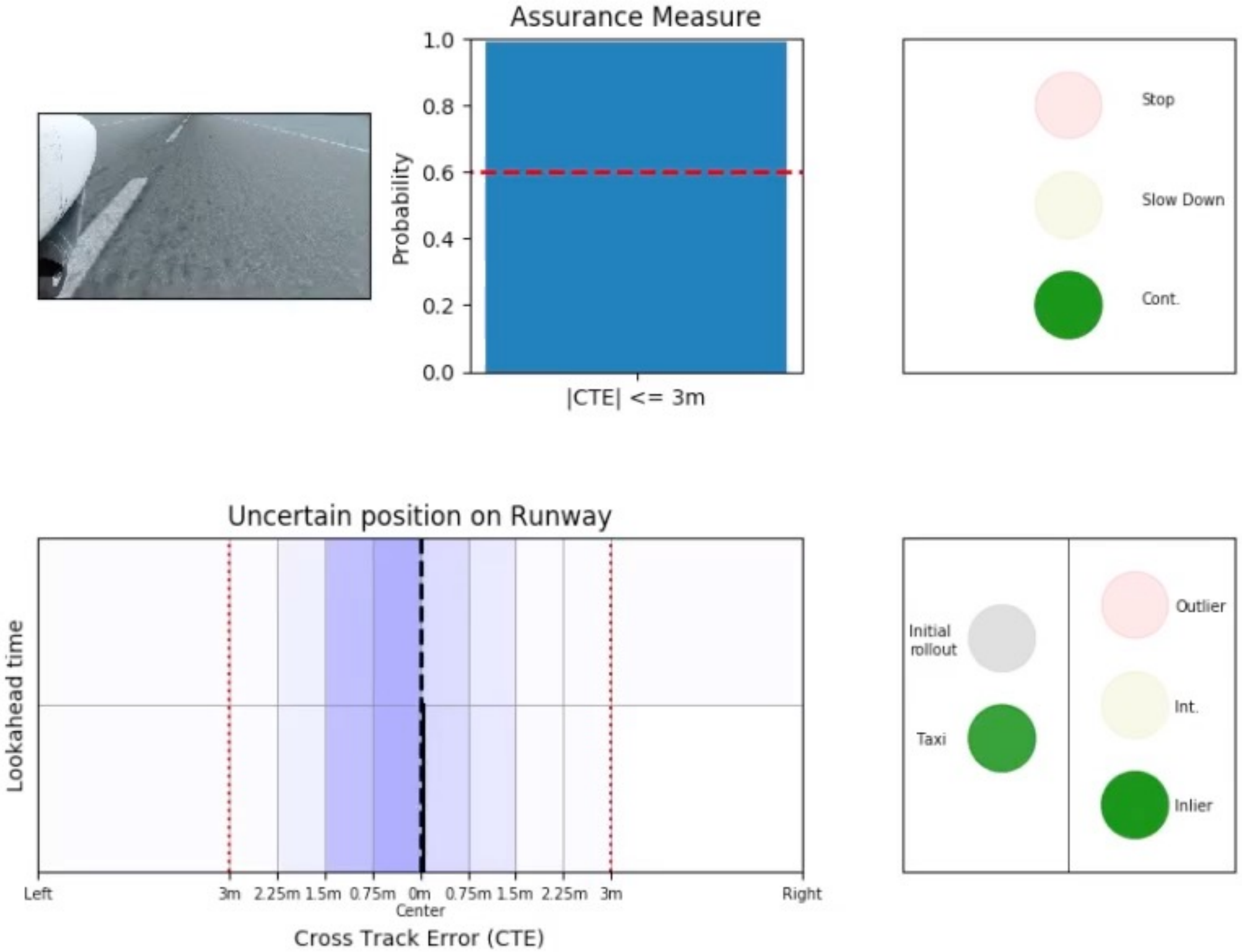


Taxiing requirement: $\pm x$ meters from centerline with 2σ confidence

- TaxiNet System
  - Autonomy Pipeline
    - Wing mounted camera
    - Deep convolutional neural network for perception
      - Provides estimate of cross-track error and heading error from camera images
    - Controller to actuate/steer aircraft
  - Safety
    - Avoid lateral runway excursion
      - Do not exceed allowed lateral offset
  - Performance
    - Do not stop too often
    - Follow centerline within allowed lateral offset for duration of taxiing

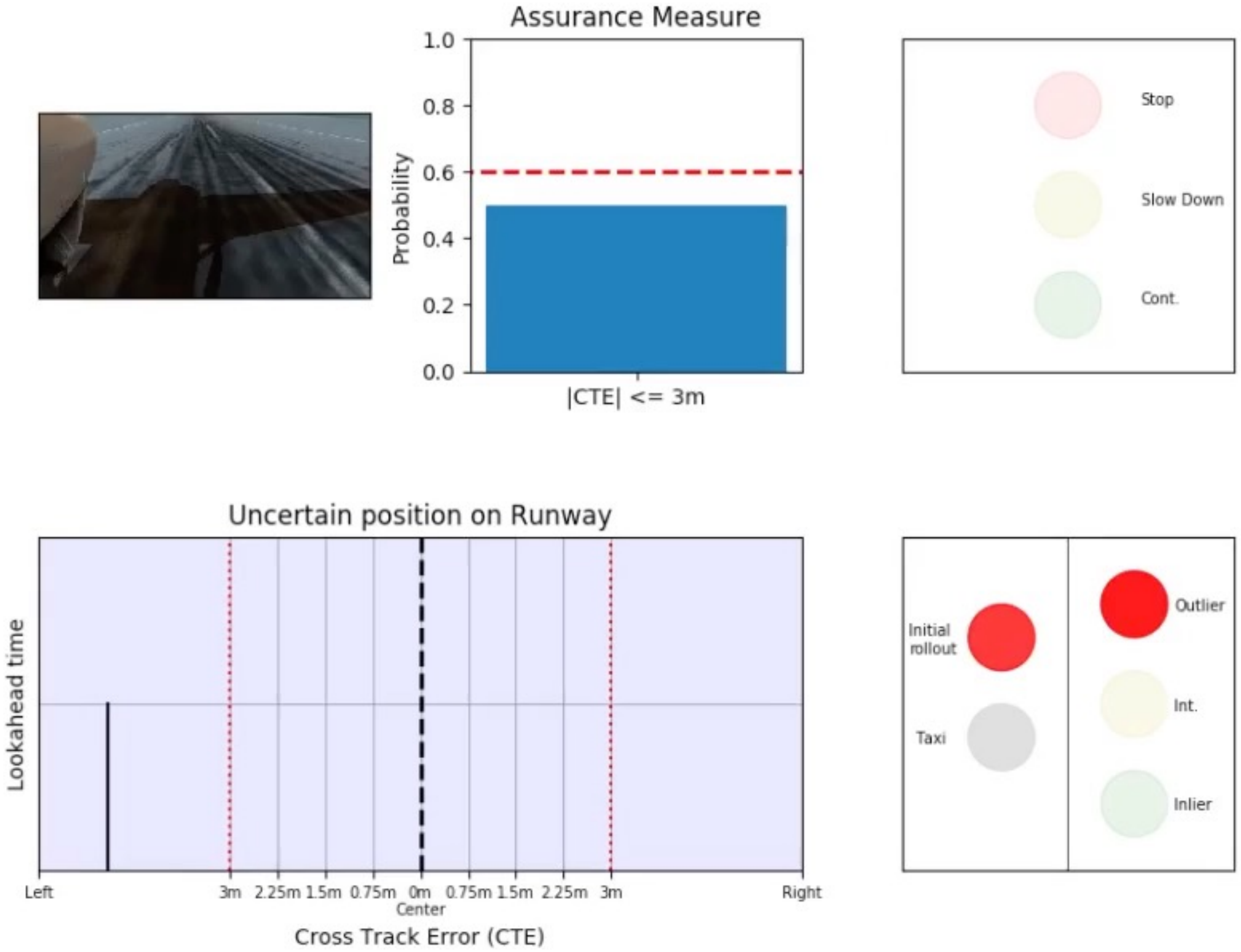
Runtime Risk Assessment

- Trained in clear and overcast weather conditions
  - LEC follows centerline
- Anomaly:
  - Runway intersection without centerline
- Assurance visualization
  - Forecast of uncertainty (confidence) in true CTE
  - Pr(Offset Violation)
  - Input anomaly detection
  - Contingency action

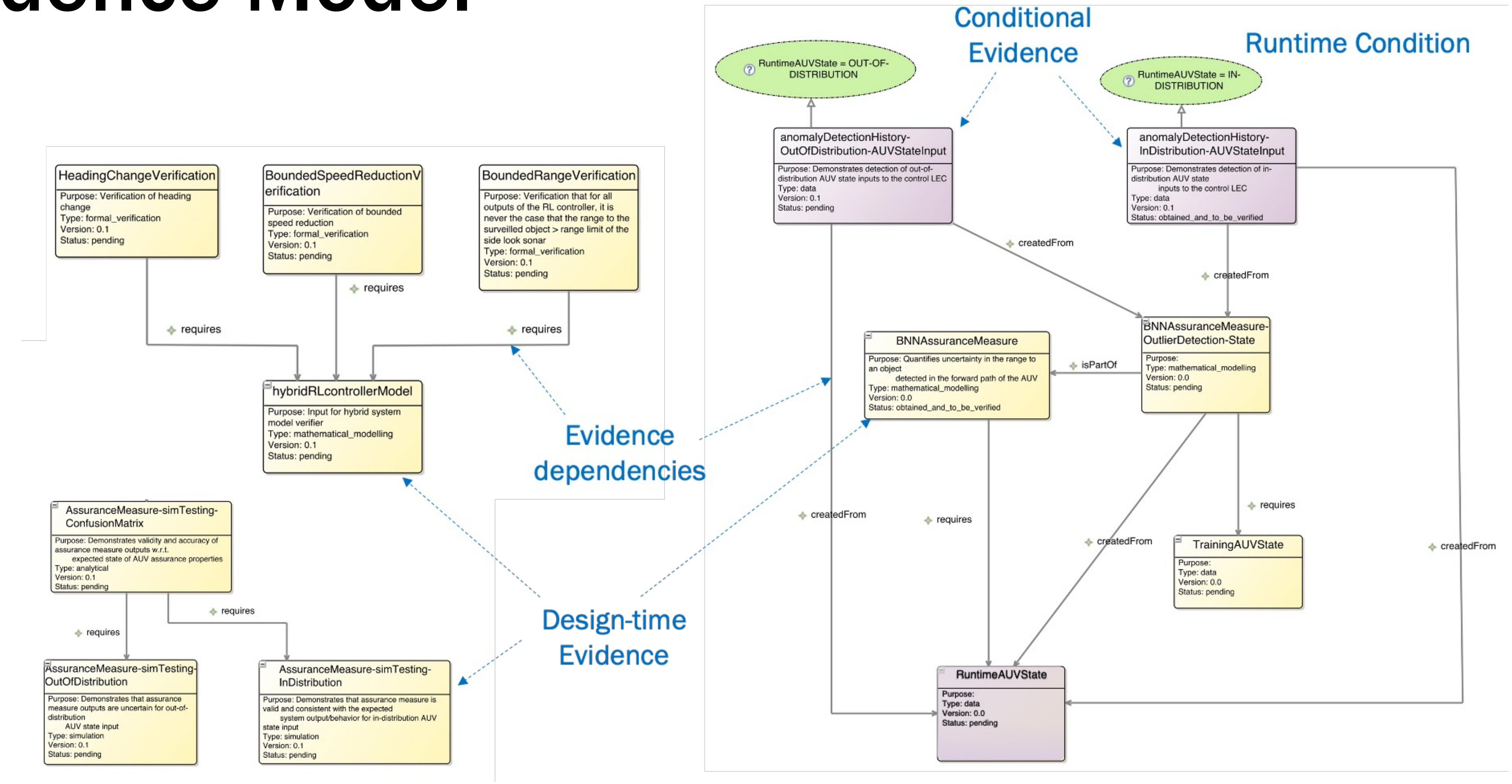


Runtime Risk Assessment

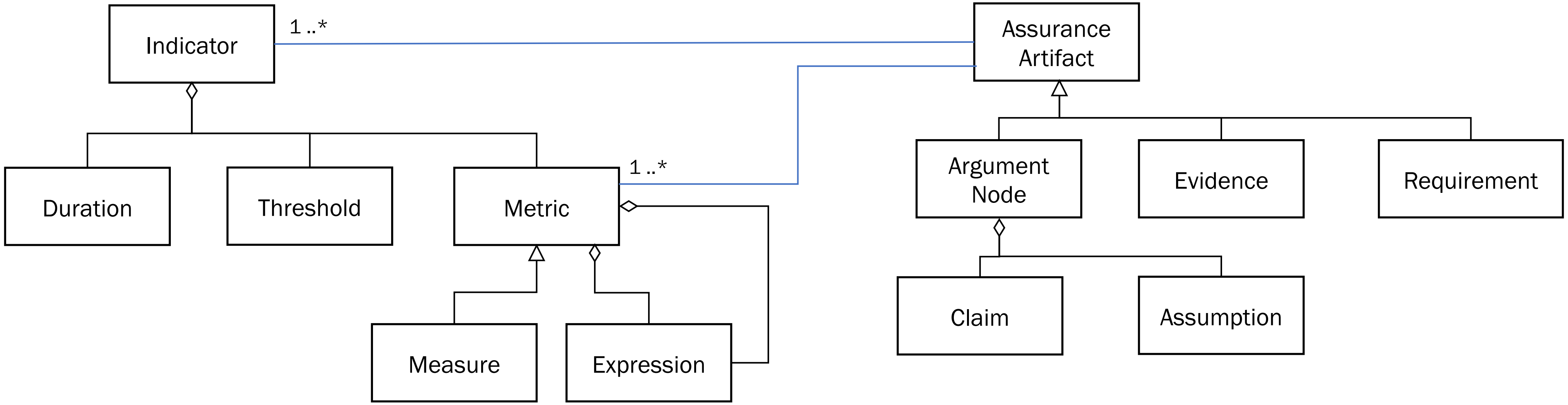
- Trained in clear and overcast conditions
- Anomalies:
  - LEC veering off centerline
  - Obscured centerline
  - Shadows
- Assurance visualization
  - Forecast of uncertainty (confidence) in true CTE
  - Pr(Offset Violation)
  - Input anomaly detection
  - Contingency action



Evidence Model



Metamodel



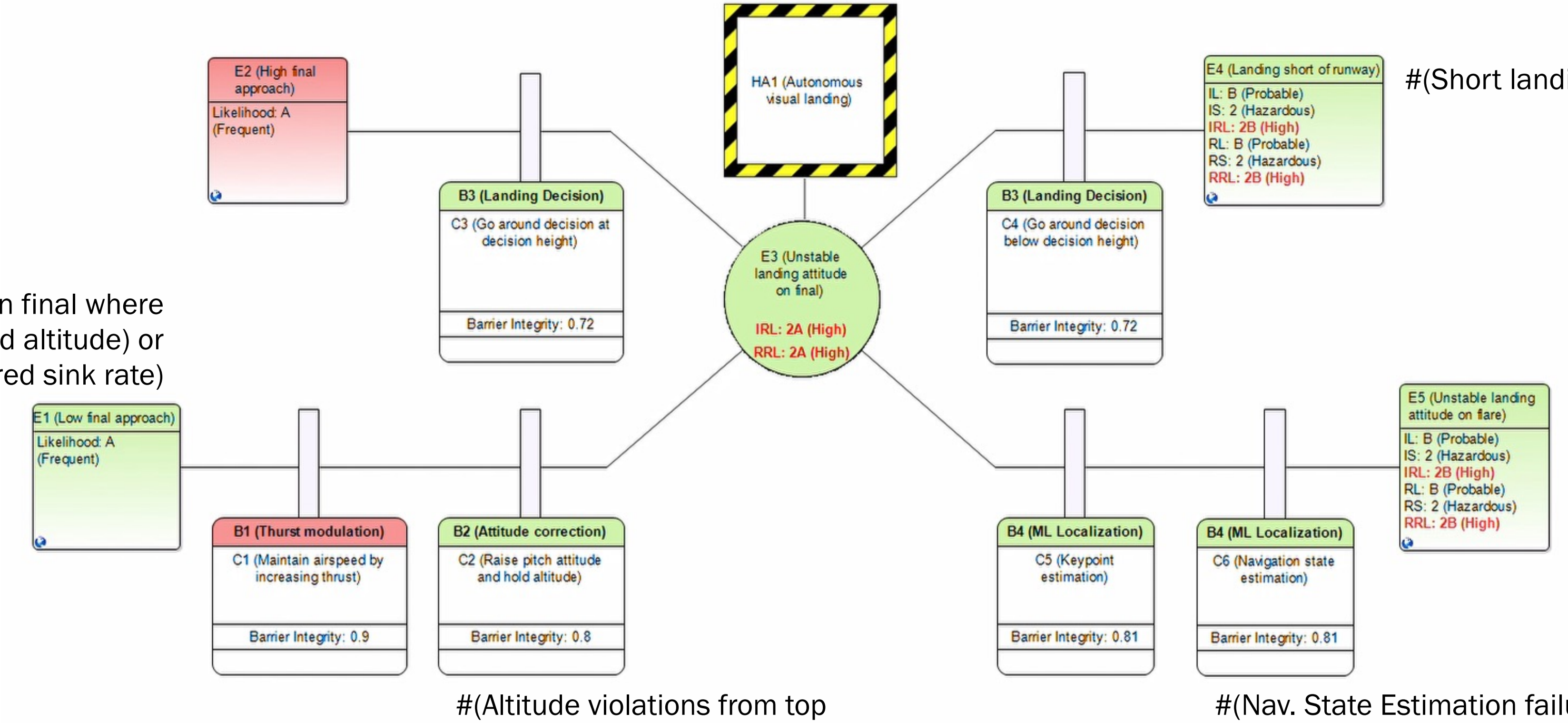
- Measures: Directly observable parameters of the system or environment
- Metrics: Computed value based on measures and other metrics
- Indicator: Target value that a metric reaches in a given duration
  - Safety performance indicators

Operational Safety Management System

- Safety Management System
  - The formal, top-down, organization-wide approach to managing safety risk and assuring the effectiveness of safety risk controls. It includes systematic procedures, practices, and policies for the management of safety risk. (FAA Order 8000.369)

- Operational SMS
  - Collection of indicators, each of which is associated with an assurance artifact
  - Complies with structure of dynamic evidence model

Defining Metrics



#(Short landings per 10K landings)

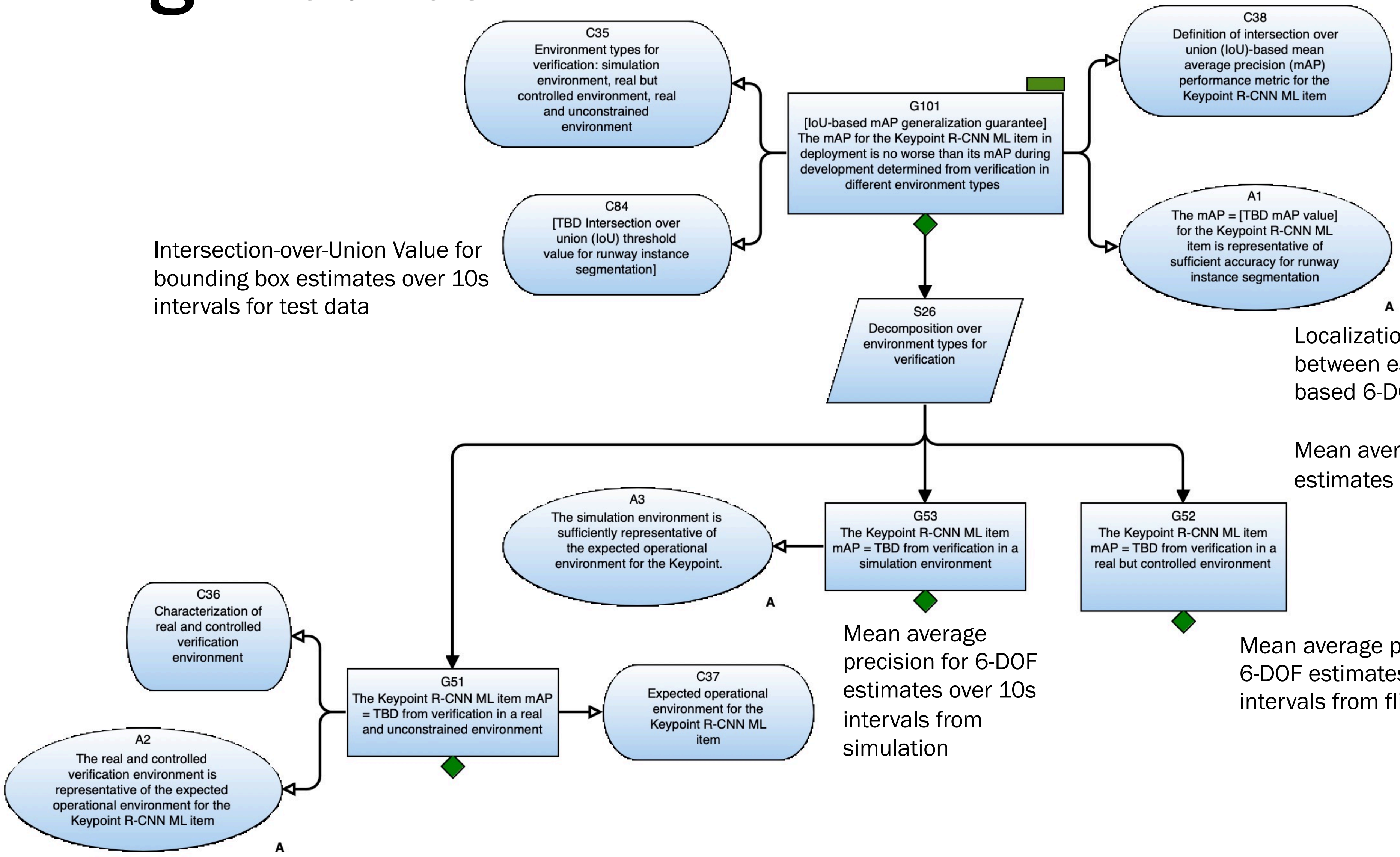
#(Approach on final where (altitude < required altitude) or (sink rate < required sink rate)

#((Flares where 6-DOF parameters are outside required ranges) per 10K landings)

#(Altitude violations from top of descent till entry on final in 10s intervals)

#(Nav. State Estimation failure conditions per 10s intervals over descent duration)

Defining Metrics



Intersection-over-Union Value for bounding box estimates over 10s intervals for test data

Localization accuracy: Deviation between estimated 6-DOF and IRS based 6-DOF < Threshold

Mean average precision over 6-DOF estimates over 10s intervals

Mean average precision for 6-DOF estimates over 10s intervals from simulation

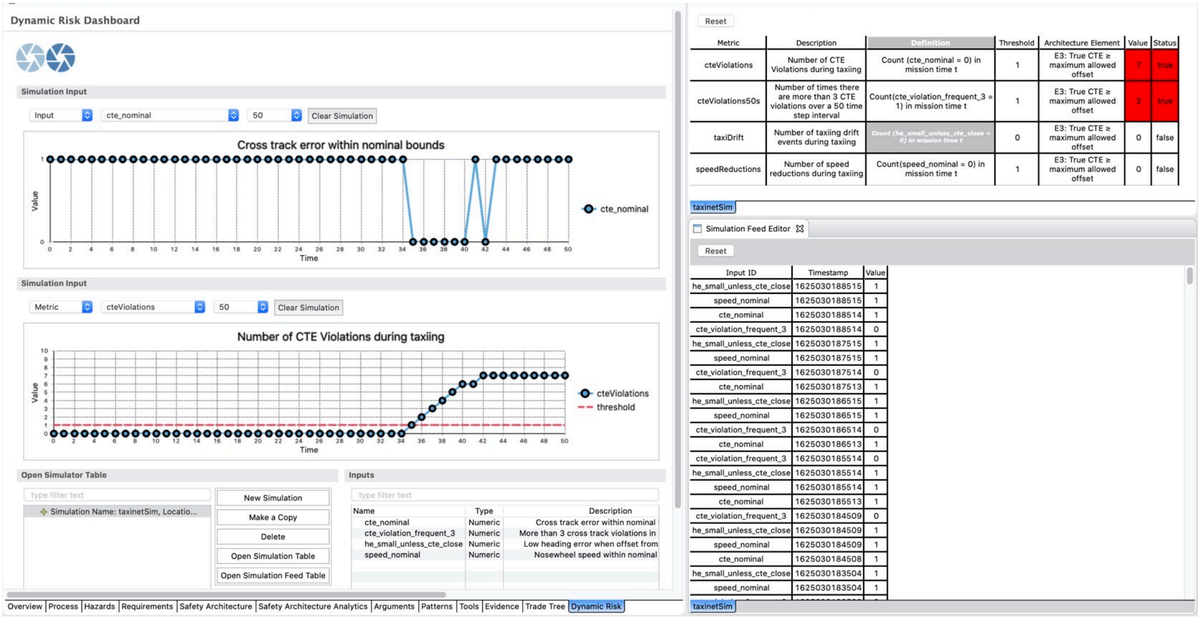
Mean average precision for 6-DOF estimates over 10s intervals from flight tests

Implementation in AdvoCATE

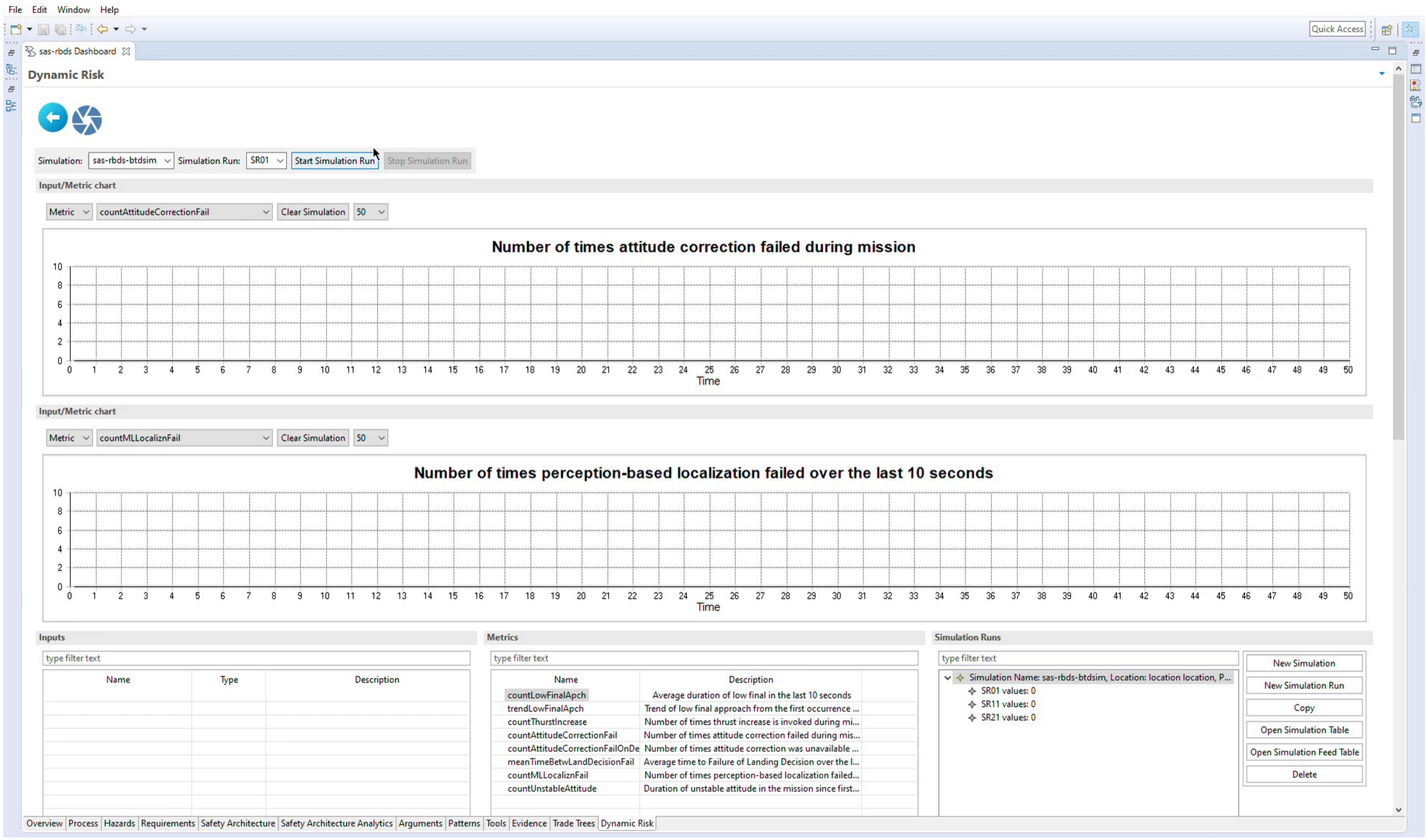
Example metrics for autonomous taxiing example

Metric	Description	Definition	Threshold	Assurance Artifact	Value	Status
oodImgIp	Number of outlier image inputs to TaxiNet during taxiing	Count (imgOutlier = TRUE) in mission time t	0	E11: Outlier image input to TaxiNet	1	true
incorrCTEcomp	Number of incorrect CTE computations during taxiing	Count (taxinetError = TRUE) in mission time t	0	E10: Inaccurate TaxiNet computation of CTE	0	false
incorrCTEop	Number of incorrect CTE estimates sent to the controller during taxiing	Count (cteEstimateError = TRUE) in mission time t	0	E7: Incorrect CTE estimate sent to controller	1	true
cteViolations	Number of CTE violations during taxiing	Count (cteViolation = TRUE) in mission time t	2	E3: True CTE ≥ maximum allowed offset	3	true
cteViolations5s	Number of CTE violations over a 5 second interval	Count (cteViolation = TRUE) from first occurrence for the next 5 seconds	1	E3: True CTE ≥ maximum allowed offset	0	false
pidErr	Number of PID controller errors during taxiing	Count (pidError = TRUE) in mission time t	0	E5: PID controller error	2	true
locLatTraj	Number of loss of control of lateral trajectory events during taxiing	Count (loc = TRUE) in mission time t	0	E4: Loss of control of lateral trajectory	2	true
locLatTraj5s	Number of loss of control of lateral trajectory events for the next 5 seconds	Count (loc = TRUE) from first occurrence for the next 5 seconds	0	E4: Loss of control of lateral trajectory	0	false
taxiDrift	Number of taxiing drift events during taxiing	Count (taxiDrift = TRUE) in mission time t	2	E1: True CTE >> 0.5 (Runway width)	2	false
latExcursions	Number of lateral runway excursions during taxiing	Count (excursion = TRUE) in mission time t	0	E2: Lateral runway excursion	0	false
latExcursionsT	Number of lateral runway excursions over all taxiing runs	Count (excursion = TRUE) in operational lifetime T	1	E2: Lateral runway excursion	0	false
cmInvok	Number of contingency manager invocations during taxiing	Count (brake = TRUE) in mission time t	0	B1: Contingency management	1	true
cmErr	Number of contingency manager failures during taxiing	Count {[(loc = TRUE) AND (brake = FALSE)] OR [(loc = FALSE) AND (brake = TRUE)]} in mission time t	0	B1: Contingency management	2	true
speedReductions	Number of speed reductions during taxiing	Count [(cteViolation = TRUE) AND (brake = TRUE)] in mission time t	2	E3: True CTE ≥ maximum allowed offset	1	false

Metrics Visualization, connected to Simulations



Implementation in AdvoCATE – Metrics Visualization



Dynamic Argument Update

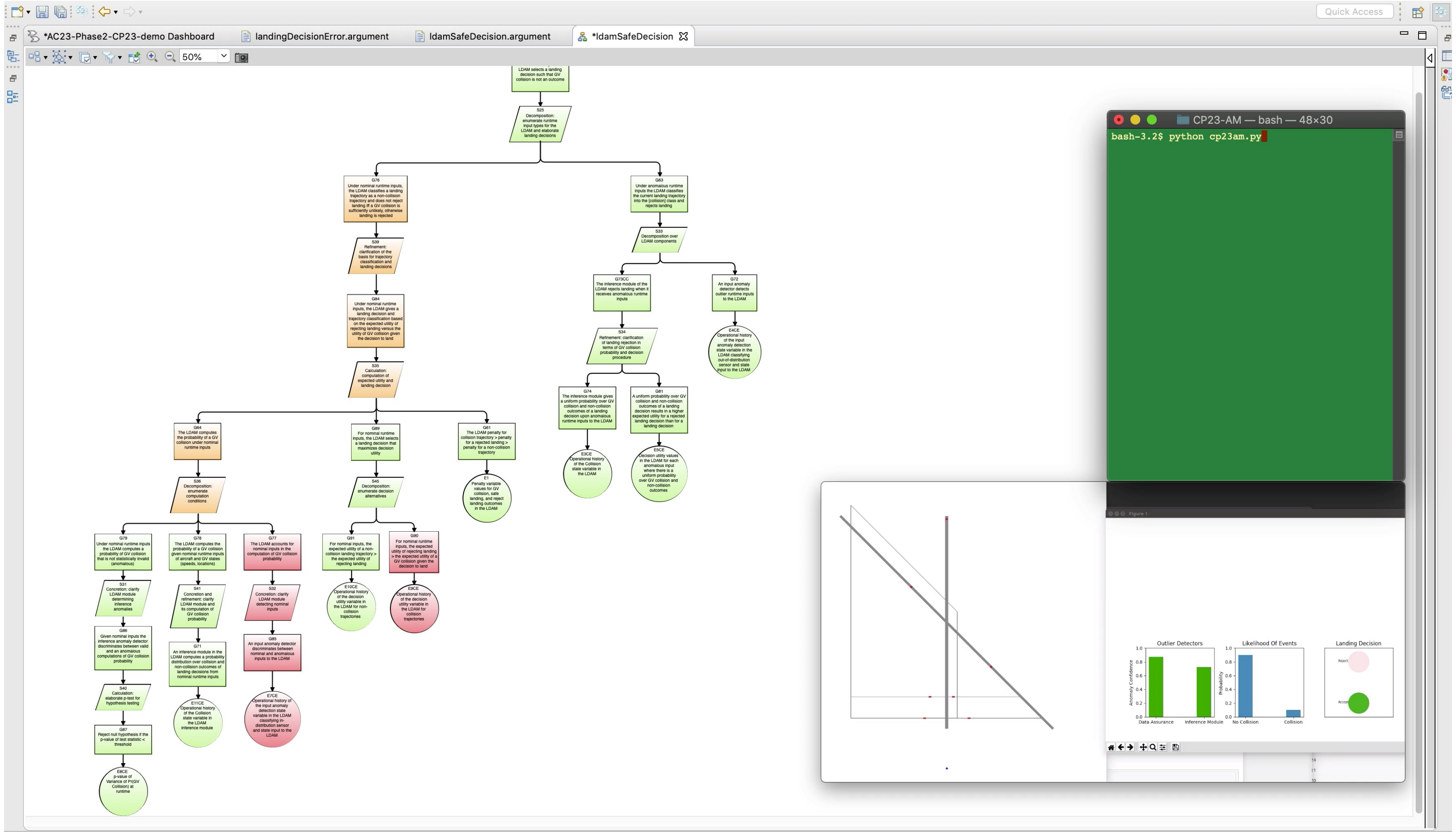
- Linking arguments and argument update to
  - Assurance measures
  - Dynamic metrics
- Deciding what to monitor and update
  - *Operational* claims (and evidence) vs. *Design-time* claims and evidence
  - Reflected in assurance architecture as events, escalations, and mitigations
- Conditional evidence specification on solution nodes using DSL
  - e.g., monitored signal received for last 3 time steps
    - `%x% != null for 3`
  - e.g., AUV control LEC inputs for current time step are out of distribution
    - State of random variable in anomaly detection component of assurance measure < threshold
      `%non_conf% < 20`

Dynamic Argument Update

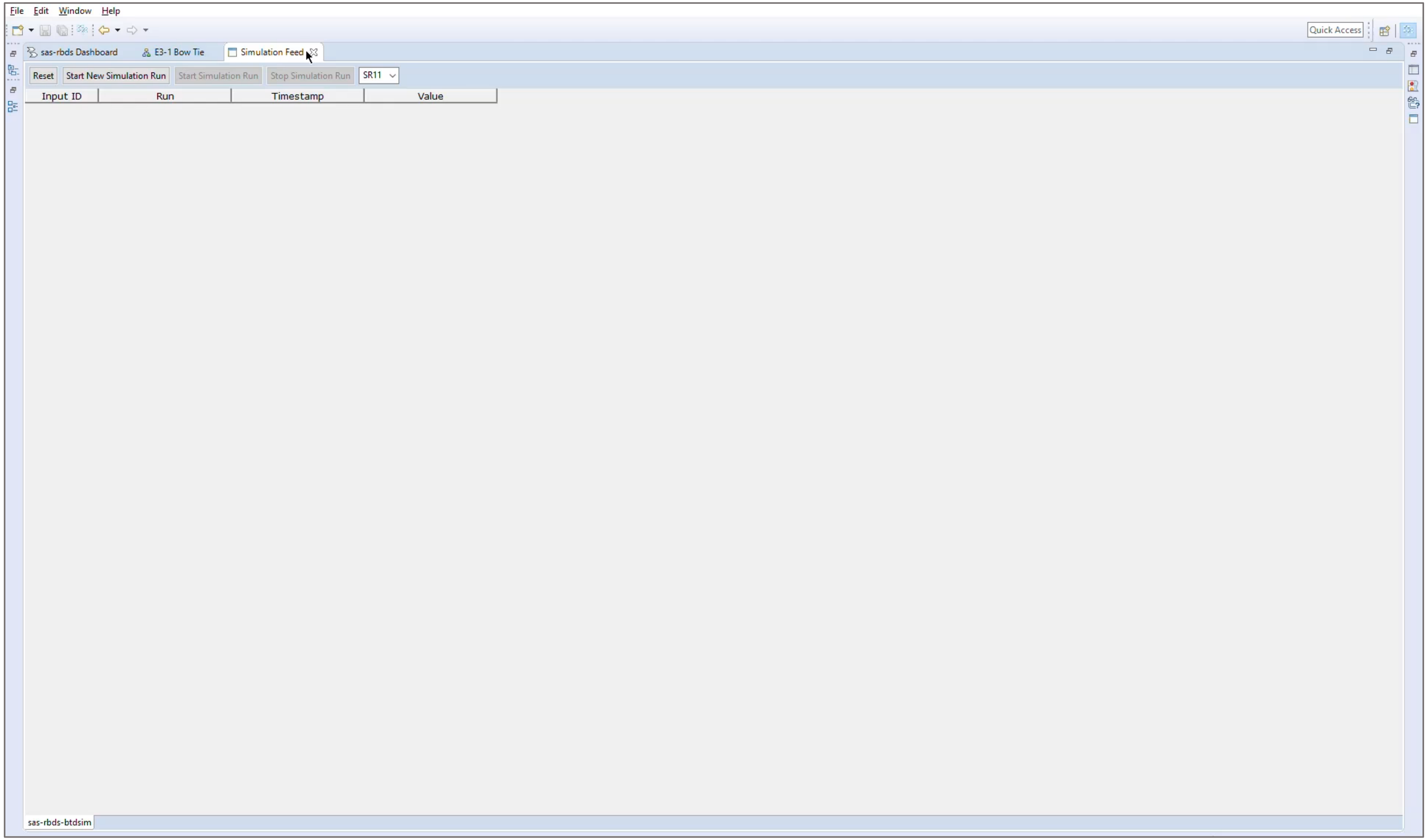
- Confidence visualization
  - Stoplight node coloring
    - Red: Very Low / No confidence
    - Green: Very High / Full confidence
    - Orange: Uncertain
- Confidence propagation as color propagation rules
  - Apply node coloring to conditional evidence based on monitored data
  - Parent node color based on conditional evidence, and argument structure
- Simple propagation rules
  - AND structure: Goal → Strategy → (Sub-goals)
    - Parent goal is green only if all child goals are green
    - Parent goal is orange if k/n child goals are orange or green, and none of the n-k child goals are red.
    - Parent goal is red if k/n child goals are red and none of the n-k child goals are green (permissive)
    - Parent goal is red if any child goal is red (strict)
- Alternative visualizations and rules
  - Confidence categories / intervals
  - Colors spectrum ∝ uncertainty (probability density) / Belief mass
  - e.g., Evidential (Dempster-Shafer) theory
  - Subjective logic

Implementation in AdvoCATE – Dynamic Argument Update

- Assured visual landing



Implementation in AdvoCATE – Dynamic Bow Ties



Dynamic Update to Risk Assessment

- Pr(c) = p : Allocated event probability
  - p is an expected upper bound
  - Equivalent to m observations of event C in N missions or simulation runs
  - Observations of C could be direct or inferred via some metric M(C)
    - M(C) = m
    - Pr(c) = $p = m/N$
- Let metric threshold T = t
  - Number of additional observations of the state of event C to be made in k subsequent missions / simulation runs
- p ideally constant across all missions
  - $m/N = (m + t)/(N + k) \rightarrow t = k(m/N) \rightarrow t = k\,p$
- Update rules for events
  - No updates necessary when observed M(C) ≤ Nearest integer(t)
  - Otherwise
    - Updated Pr(c) = $p' = (m + t)/(N + k)$
    - Updated T for k' subsequent runs = $t' = k'\,p'$
- Similarly for barriers B_i
- Estimate residual risk for Pr(e) based on updates to Pr(c) and Pr(b_i)

Risk-based Decision Support

- $Pr(c)$ and $Pr(b_i)$ unchanged or decrease
  - $Pr(e)$ unchanged or decreases
    - Validates risk assessment
    - Design may be more conservative than necessary
  - $Pr(e)$ increases but < TLOS
    - Design is incomplete
  - $Pr(e)$ increase > TLOS
    - Design is unsafe, cease operation
- $Pr(c)$ and $Pr(b_i)$ both increase
  - $Pr(e)$ increases but < TLOS
    - Either improve $Pr(b_i)$ to reduce $Pr(e)$ to previous level
    - Extent of improvement determined from re-allocation of safety target
    - Or Accept risk
  - $Pr(e)$ increases > TLOS
    - Design likely unsafe, cease operation
    - Reassess risk and introduce new mitigations / re-design
- $Pr(c)$ unchanged, $Pr(b_i)$ changes
  - Some barriers underperform, some perform as expected, others perform better than expected
  - $Pr(e)$ increases but < TLOS
    - Improve underperforming barriers
    - Re-allocate safety targets fixing integrities of performant barriers
  - $Pr(e)$ increases > TLOS
    - Design likely unsafe; Reassess risk and introduce new mitigations / re-design
    - Cease operation
- $Pr(c)$ increases, $Pr(b_i)$ unchanged
  - $Pr(e)$ increases but < TLOS
    - Either improve $Pr(b_i)$ to reduce $Pr(e)$ to previous level, or accept risk
    - Extent of improvement determined from re-allocation of safety target
  - $Pr(e)$ increases > TLOS
    - Design likely unsafe; Reassess risk and introduce new mitigations / re-design

Next Steps

- Consistency of metrics
  - Horizontal: w.r.t. TLOS
  - Vertical: refinement
- Defining update/change process based on observations
  - System Update
    - Changes to hazard mitigations, safety performance requirements
  - System – Assurance Case Interface
    - To metrics/measures/indicators
    - Associated monitors
    - Corresponding safety mitigations / contingency management mechanisms
  - Assurance Case
    - Record evidence of effectiveness
    - Changes to safety architecture, argument, and other assurance artifacts

Conclusions and Extensions

- Dynamic assurance cases
  - Closing the loop between design and operations
    - Performance measures
    - Assurance measures
    - Record design decisions
    - Update assurance case (arg + arch)
- Map decisions into updated assurance case
  - ALARP, ASARP
- Extend safety model with
  - Barrier/control dependencies
  - Repair of barriers (replace, retire, modify)
- Further automation
  - Monte Carlo simulation to generate test cases for assurance case
  - Generate structure of monitor model from safety architecture
  - Scenario generation → learn model parameters from sim
  - Training workflows
  - Generate field test plans
- Future application to wildfire mitigation using UAS