Towards an Ontological Basis for Aviation Safety Cases

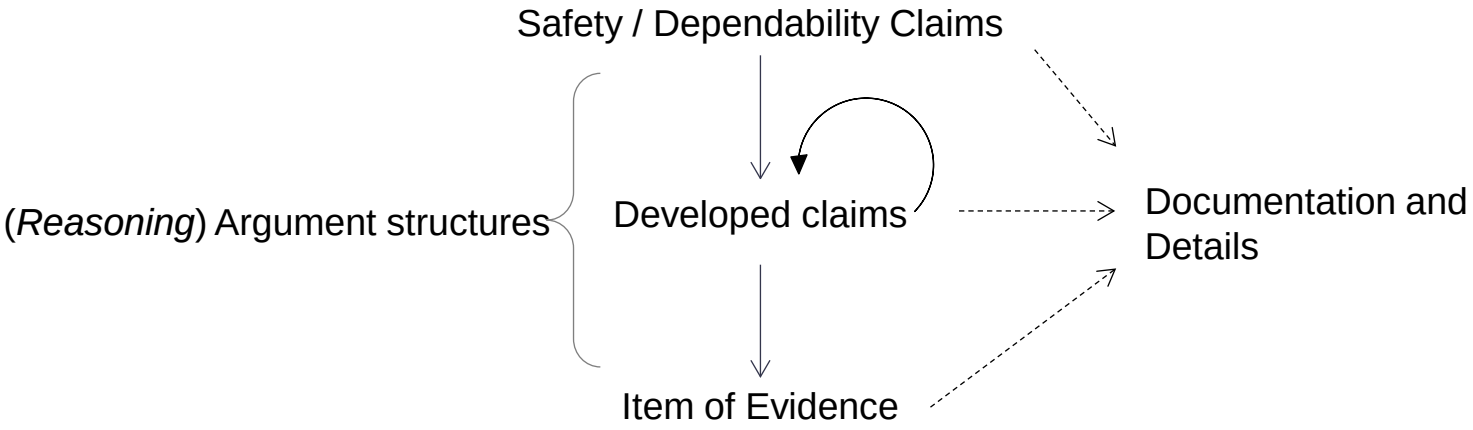
Ewen Denney and Ganesh Pai

SGT / NASA Ames Research Center

ewen.denney@nasa.gov

Safety / Assurance Case

- What we mean by “Safety / Assurance Case”
  - A structured argument
  - Linking **specific claims about safety / dependability** to a **body of evidence** that justifies these claims in a clear, convincing and comprehensive way
  - Assurance that a given system is acceptably safe / meets its assurance requirements in a specified environment



Explicit justification of how evidence generated from engineering processes (and from compliance with regulations / standards) supports claims about software and system safety.

Safety Cases in Aviation

- ICAO Guidance Material for Building a Safety Case for ADS-B separation service, May 2011
  - "A safety case is a document which provides substantial evidence that the system to which it pertains meets its safety objectives"
  - "... An explicit documentation of a safety-critical system, its corresponding safety objectives, and the associated safety risk assessment and risk management of the system, at appropriate milestones in the life of the system".

Safety Cases in Aviation

- FAA
  - Order 8900.1 Flight Standards Information Management System, Vol. 16, UAS, Ch. 7, SRM, Safety Case Template
    - "Core" content
      - Environment (airspace system) description
      - System description and system change description
      - Airworthiness description of affected items
      - Aircraft capabilities and flight data
      - Accident / incident data
      - Hazard analysis and details of risk analysis, risk assessment, and risk control
      - Emergency and contingency procedures
      - Pilot / crew roles and responsibilities
  - Safety Risk Management Plan
    - Hazard tracking
  - Required in certain circumstances: e.g., Alternative Means of Compliance with See-and-Avoid (14 CFR 91.111, 91.113, 91.115)

Example

MIZOPEX Ground-based Sense and Avoid (GBSAA)

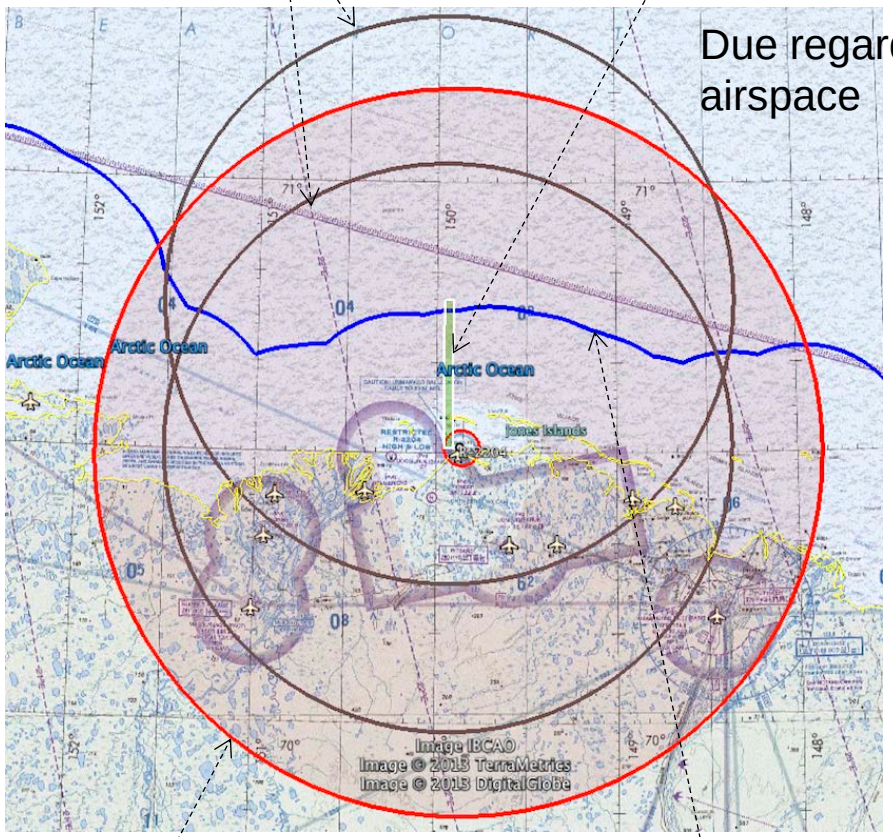
- Performing Earth Science measurements in the Arctic Ice
  - Off the coast of Alaska (Oliktok Point)
  - Satellite-based solution was too expensive
  - Use airborne instruments on UAS
    - Two classes of small UAS
    - NASA SIERRA; University of Alaska's Boeing Insitu ScanEagle
  - Too dangerous for visual observers
    - So use ground-based air defense RADAR for "sense-and-avoid"

- Considered an alternative means of compliance (AMOC) by the FAA
  - Hard requirement to submit a safety case for approval of operations by means of a Certificate of Authorization (COA)
  - Use N 8900.207, FAA National Policy Document on UAS operational approval guidance (now replaced by N 8900.227)
  - Our role
    - Create an operational safety case for this AMOC

MIZOPEX GBSAA Concept

SIERRA UAV

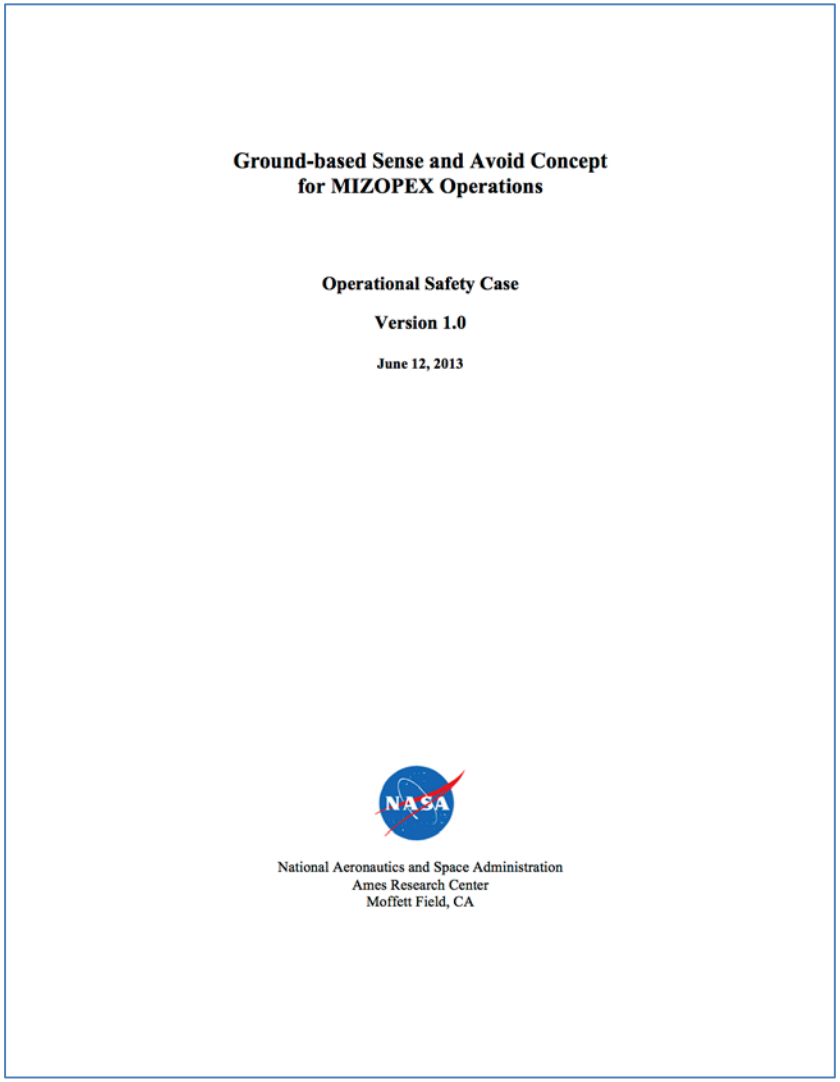
Air Defense RADAR for monitoring and airspace deconfliction



MIZOPEX GBSAA Hazard Analysis

- GBSAA Hazard
  - Known / unknown state of the GBSAA system (which may / may not be a deviation from its required operational state)
  - One or more known / unknown classes of *environmental conditions*
  - Combinations in different *flight phases*
  - Examples
    - Loss of RADAR system to detect air traffic in the surveillance volume, *during outbound transit* when *surveillance volume previously all clear*
    - GBSAA functioning as required, *with non cooperative aircraft in the threat volume not covered by the surveillance volume on an intercept flight path*, when *UA is outbound in the transit corridor*.
  - 5 known states, 8 flight phases, 3 classes of environmental conditions ~ 26 cases leading to potential mid-air collision
  - Collision with terrain managed through range safety

MIZOPEX GBSAA Operational Safety Case



Ground-based Sense and Avoid Concept for MIZOPEX Operations

Operational Safety Case

Version 1.0

June 12, 2013

National Aeronautics and Space Administration
Ames Research Center
Moffett Field, CA

- Accepted by the FAA, COAs granted
  - Primarily a report
  - Explicit argumentation not required to be communicated by the regulator
  - However, we are preparing safety arguments
  - First known example of GBSAA use for civilian UAS operations in the NAS
  - First known accepted safety case for civilian UAS operations in the NAS
  - Explicitly required hazard tracking and monitoring to validate assumptions and safety case

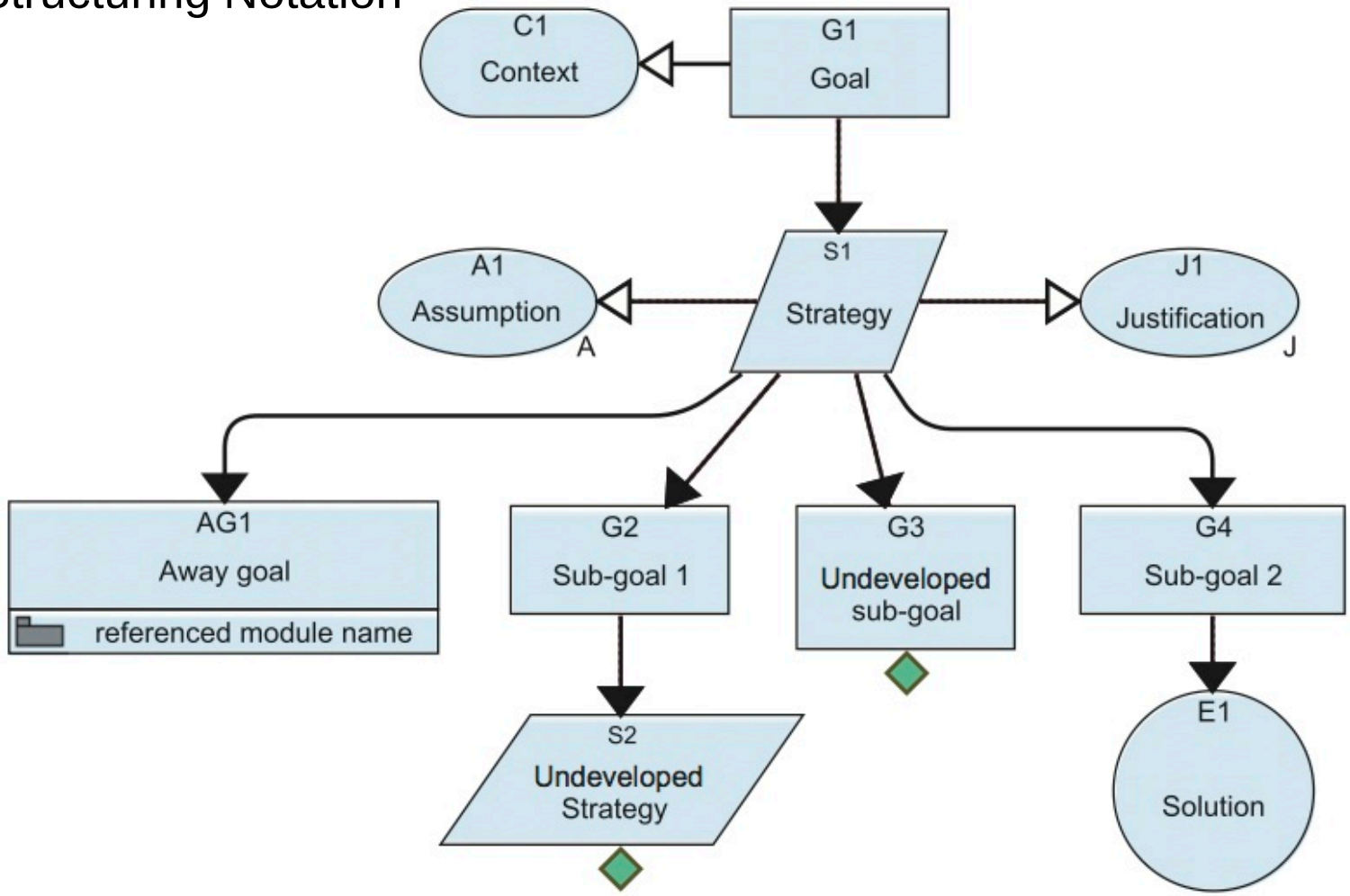
Foundations and Tool Support

Assurance Cases – Research/Tool Challenges

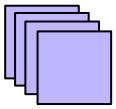
- Argument Construction
  - Interactive and/or automated
  - Integration of external sources, e.g., safety analysis, engineering analysis, requirements analysis, formal verification, ...
- Argument Evaluation
  - Verifying properties of arguments (Structural)
  - Validation of argument content against domain (Semantic)
- Argument Insight
  - Queries & Views
  - Stakeholder relevant information management
- Process Support
  - Generation of traceability matrices
  - Metrics-based evaluation
  - Confidence assessment
  - Decision making (Go / No Go)
  - Report generation

Documenting a Safety Argument

Goal Structuring Notation (GSN)

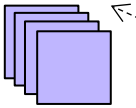


Argument Structures and Safety Cases



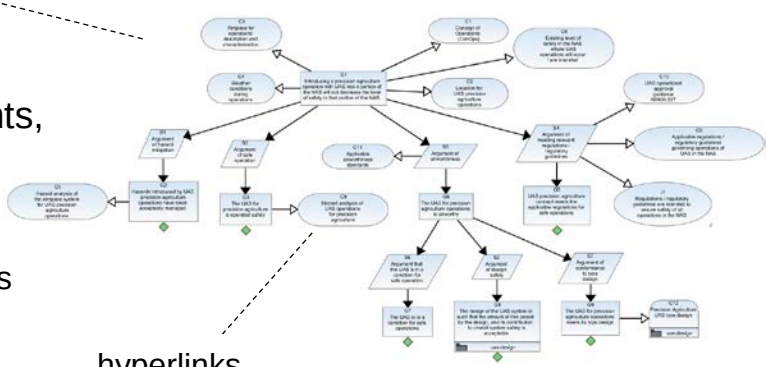
hyperlinks

External Documents
e.g., hazard logs, requirements, etc.

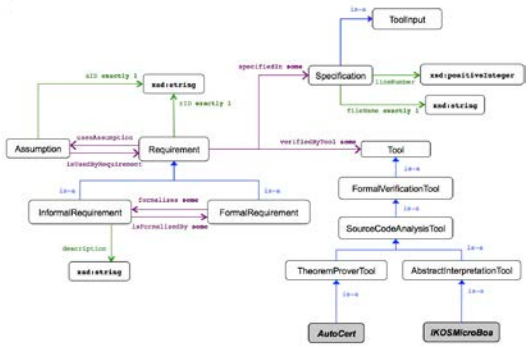


hyperlinks

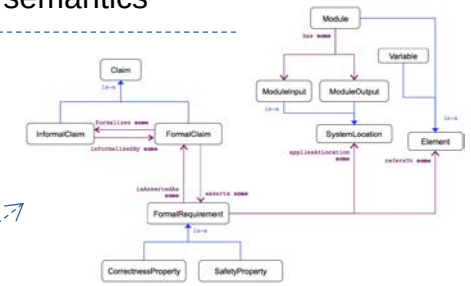
Argument Structures
e.g., in GSN
with well-formedness constraints



hyperlinks

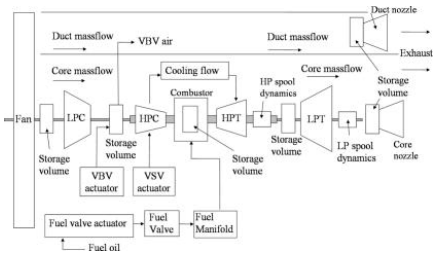


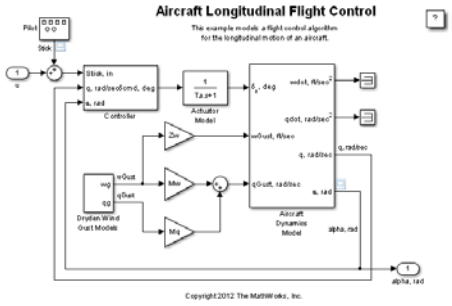
semantics



Domain model

Models / Artifacts of the System
e.g., in MATLAB / Simulink, etc.





Ontologies
e.g., in OWL
- System organization
- Regulations
- Environment / Domain, etc.

All of this constitutes the safety case

Lightweight Semantics

- Modeling domain knowledge
  - Ontologies can provide domain-specific semantics to argument structures
  - Capture as metadata associated with argument structure nodes
  - Attribute syntax

```
attribute ::= attributeName param*
param ::= String | Int | Nat | nodeID | sameNodeTypeID | goalNodeId | strategyNodeId |
          evidenceNodeId | assumptionNodeId | contextNodeId | justificationNodeId |
          contextNodeId | userDefinedEnum
```

  - Examples
    - userDefinedEnum

```
severity ::= catastrophic | hazardous | major | minor | noSafetyEffect
likelihood ::= frequent | probable | remote | extremelyRemote |
               extremelyImprobable
```

    - Attribute: risk(severity, likelihood), formalizes(sameNodeTypeID)
    - Attribute instance: risk(severity(catastrophic), likelihood(remote))
    - Parameter type synonyms: requirement == string

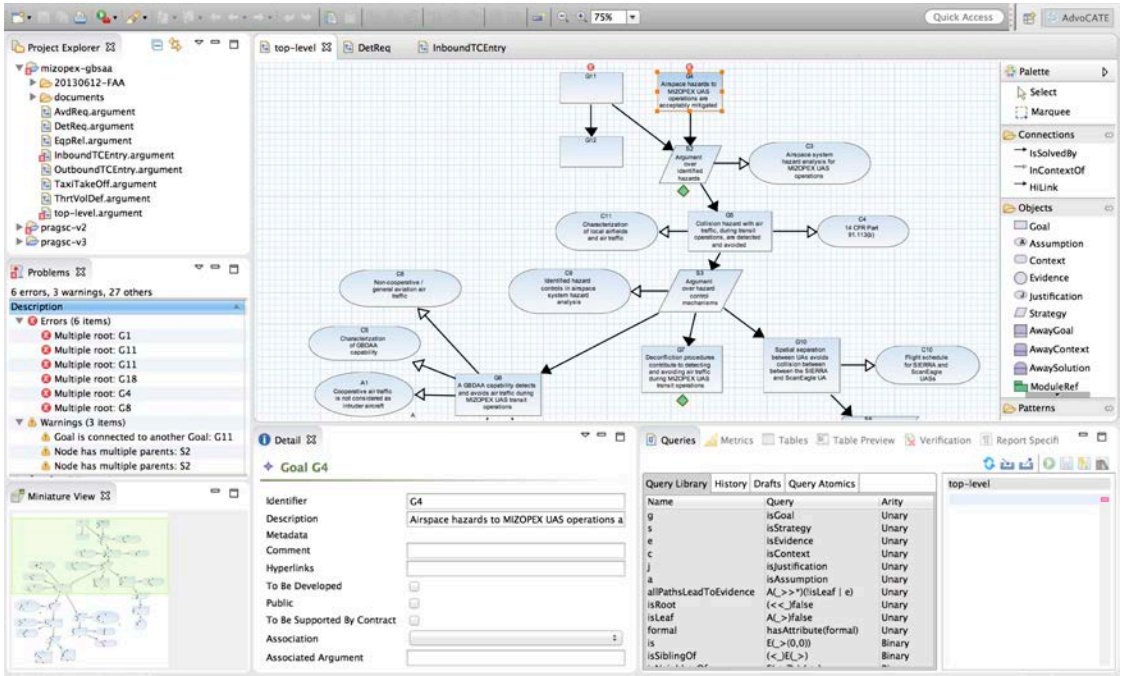
Example

requirement(id, hierarchyLevel, assuranceConcern)
formalClaim(id), informalClaim(id), hazard(id)
    id ::= int | string
    hierarchyLevel ::= highLevel | lowLevel
    assuranceConcern ::= functional | safety | reliability | availability | maintenance
requirementAppliesTo(elementLevel, elementType, element)
    elementLevel ::= system | subsystem | component | module | function | model | signal
    elementType ::= hardware | software
    element ::= aileron | elevator | flaps | propulsionBattery | avionicsBattery | actuatorBattery |
        avionics | autopilot | FMS | AP | aileronPIDController | elevatorPIDController |
        propulsion | engine | propeller | engineMotorController | actuator |
        flightComputer | wing | actuatorMotorController pilotReceiver | IMU |
references(variable)
    variable ::= aileronValue | pitchAttitude | flareAltitude | vRef | vNE | thrust | vS1
regulation(part)
    part ::= 14CFR23.73 | 14CFR23.75
risk(severity, likelihood)
    severity ::= catastrophic | hazardous | major | minor | noSafetyEffect
    likelihood ::= frequent | probable | remote | extremelyRemote | extremelyImprobable
isFormalizedBy(sameNodeTypeID)

Motivating Automation

- Maintaining consistency and supporting evolution
  - Systems and safety cases evolve
  - Keep consistent during development / in operation
- Structuring large arguments
  - Modularization
  - Hierarchisation
- Aiding stakeholder comprehension
  - Diverse stakeholders care about different things
- Supporting analysis and review
  - Assess progress, coverage, confidence
- Supporting reuse
  - Extract reusable safety artifacts

AdvoCATE: Assurance Case Automation Toolset



- Creation of safety / assurance argument
  - Hyperlinks in nodes to documents, data for evidence, context, etc.
  - Metadata on nodes: hazards, high/low requirements, risk (severity, likelihood), provenance

Vision

Safety infor**m**ation, **a**ssurance and **r**isk managemen**t**

(SMART) Dashboard

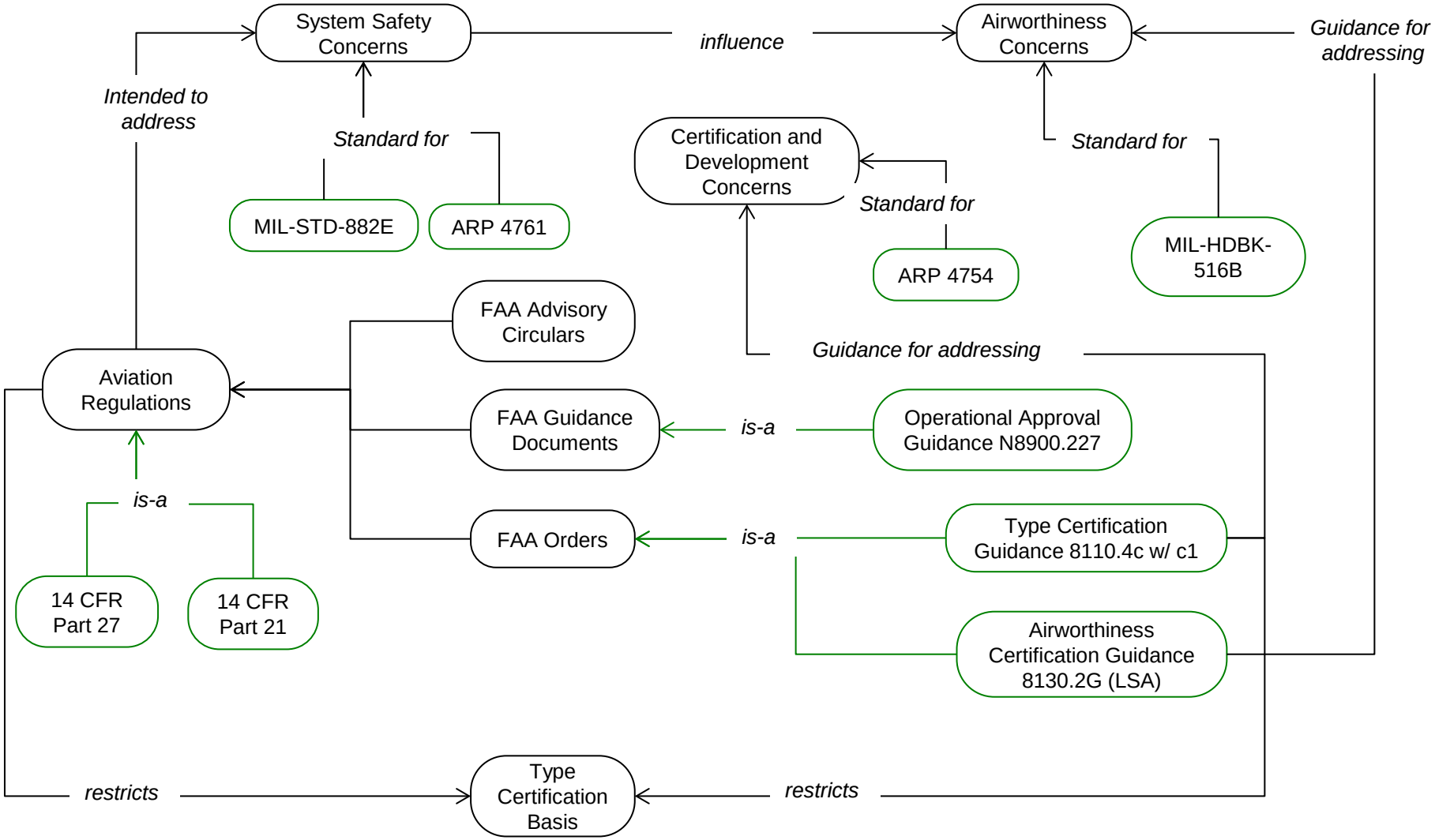
- Functionality
  - Report generation
  - Generation of to-do lists
  - Generation of traceability matrices
  - Computation of metrics
  - Queries, views
  - Verification
- Structuring
  - Patterns
  - Modules
  - Hierarchy
- Integration/generation
  - Requirements tables
  - Formal methods

Towards the Use of Ontologies

Information Sources

- Many related information sources
  - Type Certification Guidance, e.g., Order 8110.4c w/ change 1
    - Type Certification Basis
  - Airworthiness Certification Guidance, e.g., Order 8130.2G
    - FAA and Industry Guide to Product Certification
  - Existing regulations and related documentation
    - FARs, (Parts 21, 23, 27?)
    - N 8900.227, Order 8130.34B
  - Other (Standards, Guidance)
    - MIL-HDBK-516B (DoD Airworthiness Handbook)
    - ARP 4754 (Guidelines for Development of Civil A/C and Systems)
    - MIL-STD-882E, ARP 4761 (System Safety)
    - Minimal Aviation System Performance Standards (MASPS)
    - Minimal Operational Performance Standards (MOPS)
    - Technical Standard Orders (TSOs)
    - Safety Management Systems (SMSs)
    - Safety Performance Requirements (SPRs)

Example: Type Certification



From Ontologies to Arguments

- Primarily, a source of
  - Safety / airworthiness goals / classes of goals
  - Hazards
  - Strategies to develop goals
  - Evidence classes and types
- Provides the minimal outline of the goals + evidence required
  - Arguments are abstract and/or implicit in the documentation
  - Map: ontology -> argument skeleton
- Defines constraints on the argument
  - Argument ought not to contradict domain ontology
    - Unless ontology or interpretation is wrong
- Enables automation
  - Queries, views, report generation, argument validation and verification (including coverage)

Application - UTM: UAS Traffic Management

Goals: Safely Enable Routine Widespread Operations of Small UAS in:
1) Uncontrolled Airspace (Class G), and 2) throughout the NAS (at low altitudes)

Research Approach

- Develop an ontology of UTM concepts
  - A *domain model* describing the UTM system in terms of its
    - Actors, Services, Vehicle/airspace characteristics, Operations,
    - System components, Component characteristics and interrelations
  - Serves as a *formalized* and *validated* knowledge base
    - Captures agreement on terminology, concepts and relations from project stakeholders
  - Use the ontology to assist safety analysis on relevant system parts
  - Result is extensions to system or argument fragments
  - Ontology is updated
    - As new information about the system is known
    - As the argument is updated

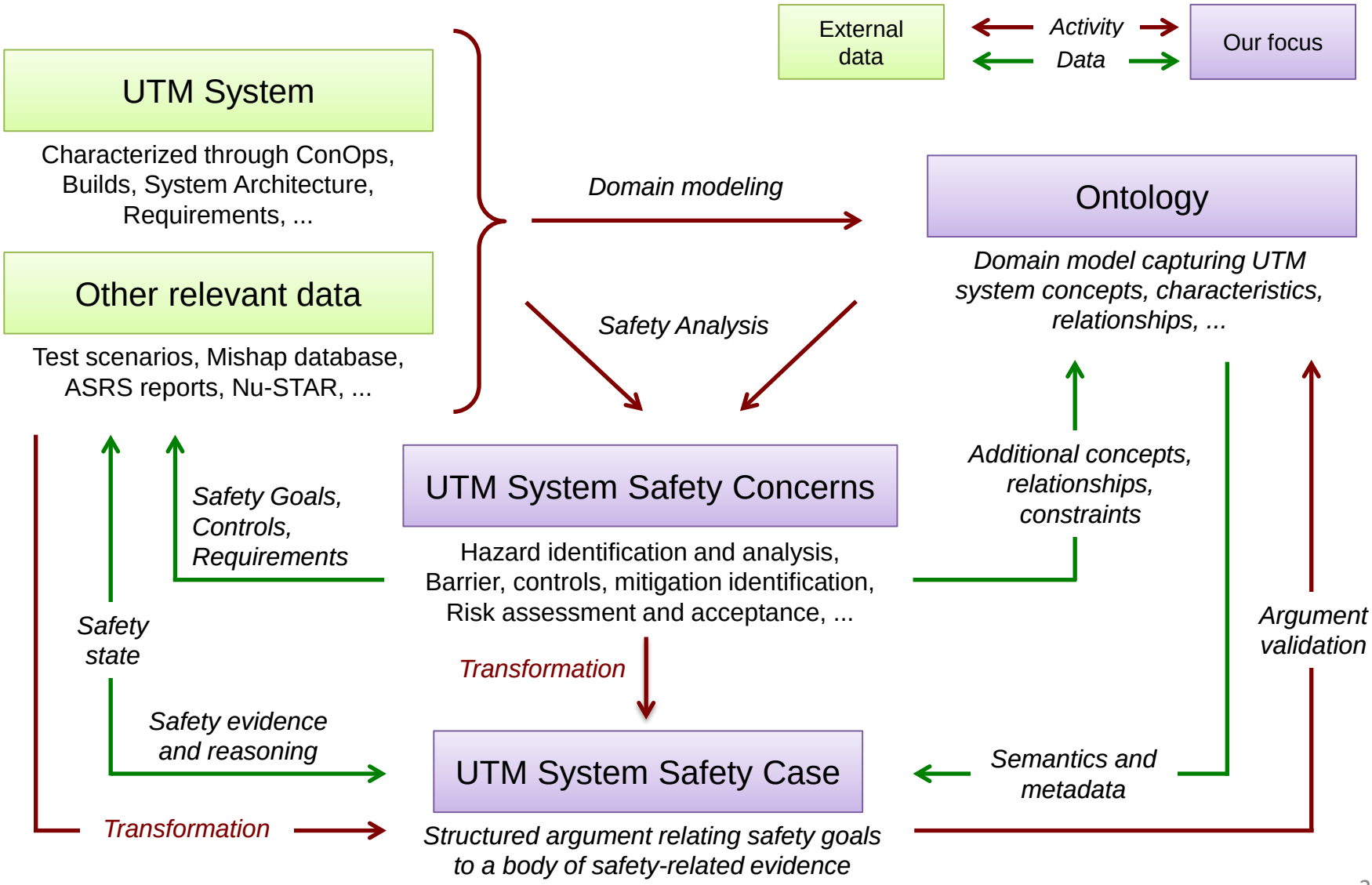
Research Approach

- Hazard Identification and Analysis
  - Primary, secondary hazards
  - Identify whether or not the existing UTM safety barriers, e.g., geo-fence, are sufficient, or if additional barriers are needed.
  - Possible hazard analysis methods: FHA, BBTA, HAZOP

- Map into a safety argument
  - The *safety analyzed* ontology
  - Engineering reasoning, and evidence (analysis, simulation, test results, etc.) produced

- Key idea
  - Future changes should be be reflected in the ontology, allowing the argument to be (automatically) checked for compliance. Track:
    - ConOps/System
    - Accident reports, mishap database, test scenarios

Research Approach



Conclusions

- Safety / Assurance cases
  - Explicitly linking safety/assurance claims to the supporting evidence via arguments
  - Explicitly highlight the rationale in processes, standards, guidelines
- Traditionally informal but ontologies can provide semantics
- Ontology-based
  - Safety analysis
  - Argument generation/update
- Aim for reusable safety artifacts
  - Argument fragments
  - Patterns
  - Domain knowledge

Questions

- Bidirectional mapping between argument and ontology
  - Concepts <-> argument nodes
  - Relations <-> inference fragments
- Other relevant work?
  - Ontologies to build on?
  - Tools?
  - Mining documents?
- Formalism: DL vs OWL vs …?
- Methodology for safety analysis / safety argumentation vs ontology creation
- Can we justify improvement?
  - Better?
  - Faster?