Architecting Safer Autonomous Aviation Systems

Jane Fenn &
Michael Wilkinson

BAE Systems

Mark Nicholson

University of York

Ganesh Pai

KBR / NASA Ames Research Center

February 7, 2023

2023 Safety-Critical Systems Symposium, York, UK

Outline

- Motivation
- Background
- Architectural patterns
  - Generic
  - For AI/ML
- Conclusions
- Future work

Motivation

- Members of SAE G-34 and EUROCAE WG-114
  - Standards committee for AI in Aviation
  - Developing AS6983 - Process Standard for Development and Certification / Approval of Aeronautical Safety-related Products Implementing AI
- Responsible for guidance on system architectures suitable for use with AI/ML
  - Format and nature of a standard inappropriate to provide guidance and instructional materials
  - Poses a gap → this paper represents firsts steps towards closing the gap
- Not just for aviation, and could be applicable in other domains
- Many people interested in developing AI have limited experience in Safety Engineering and Certification
  - Need for “entry-level” guidance
  - Capturing experience of successful solutions

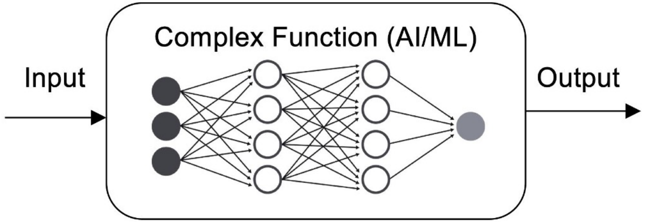
Background

- Integration of artificial intelligence (AI) and machine learning (ML) in particular poses an ongoing challenge to demonstrating a system is safe
- We have found little guidance on system level architectural selection and optimisation
  - More written at software level
  - Knowledge of architecting is implicit amongst practitioners and authors of standards/guidance documents
- Architectural approaches to dealing with untrusted or insufficiently-assured elements have been used previously in demonstrating system safety
  - e.g., COTS software components
- Architectural approaches are inevitably necessary to deal with the challenges in certification of systems using ML
  - New architectures are being proposed
  - Exploring the benefits, risks and pitfalls of various architectures captured as patterns

Background

- Many architectural approaches assume a "top down" and "greenfield" approach
  - Not realistic in practice! May be lots of constraints to consider
  - Early design-space exploration is "trial and error"
    - Involves trade-offs and optimisation
    - e.g., using Architecture Trade-Off and Analysis Method (ATAM), or Trade Trees.
- Airworthiness regulations include guidance/requirements on principles for fail-safe design
  - e.g., redundancy, warnings
- Identify some generic properties to consider in architectural selection
  - e.g., diversity, independence
- Consideration of undesired behaviour is part of making architectural choices
  - Techniques such as SHARD for guide-word directed analysis: Omission, Commission, Early, Late, Value
- Successful system architecture are often recorded as "patterns"
  - Context is aerospace so will use the terminology of commercial aerospace where systematic integrity/assurance is labelled using 'Development Assurance Levels', DAL A being highest, E lowest

Generic Architectural Patterns (1)

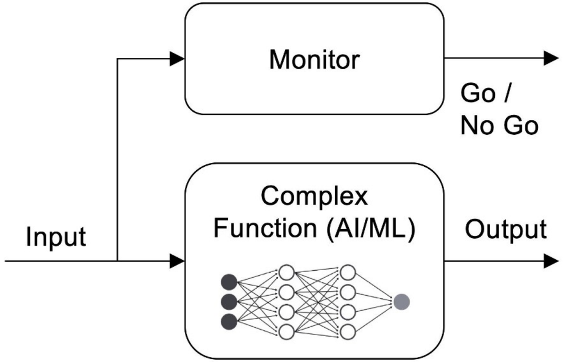


Single Channel

- Complex function inherits all probability and assurance requirements

Challenges

- No current accepted way to predict failures per operating hour of the complex function
- No current accepted (in aviation, by the regulator) approach to demonstrating high levels of assurance for ML



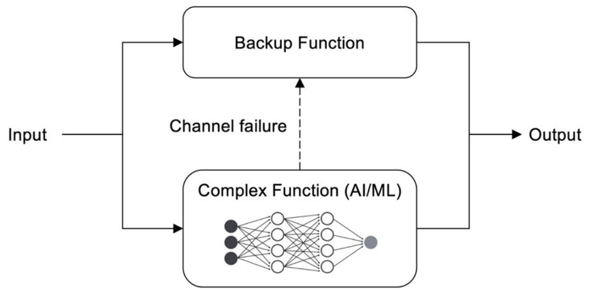
Active Monitor Parallel Design

- Predominantly for handling erroneous behaviour
- Assumes the input space for which the complex function cannot be trusted can be determined and a separate system chooses to use or not use the complex function’s output

Challenges

- Complexity of high assurance monitor if the input space for which the complex function cannot be trusted is complex

Generic Architectural Patterns (2)

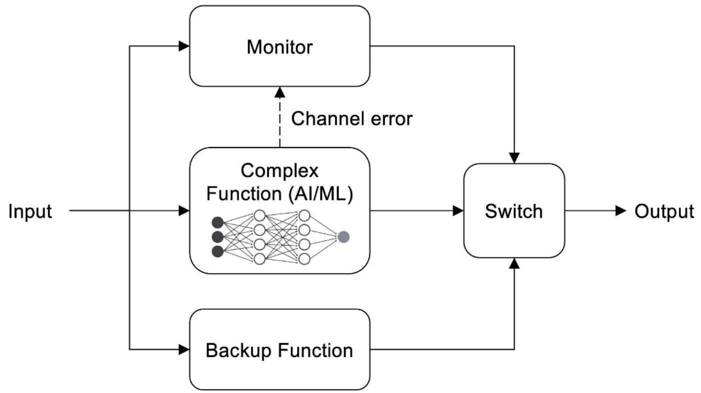


Backup Parallel

- Aids availability and dealing with “loss of function”
- Traditionally, Backup function may be a simpler, lower assurance function, but for ML, likely to be use as the higher assurance option

Challenges

- Self-diagnosis of failure/error by low assurance part
- Availability of the complex function channel, in context, needs to be assessed for safety



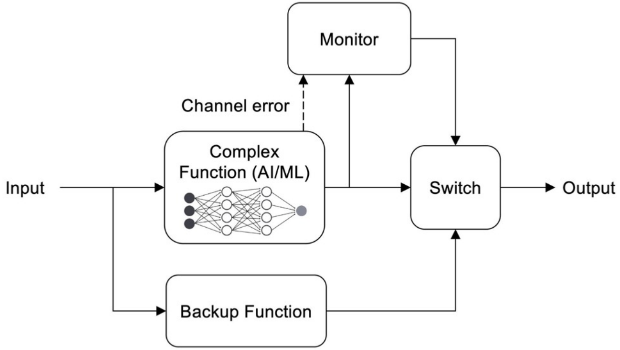
Combination of generic architectural patterns (1)

- Loss of function easier to detect externally, so focus on erroneous/incorrect function
- Trying to take advantage of higher assurance conventional backup for better availability

Challenges

- Still reliant on self-diagnosis of failure/error by low assurance part

Generic Architectural Patterns (3)

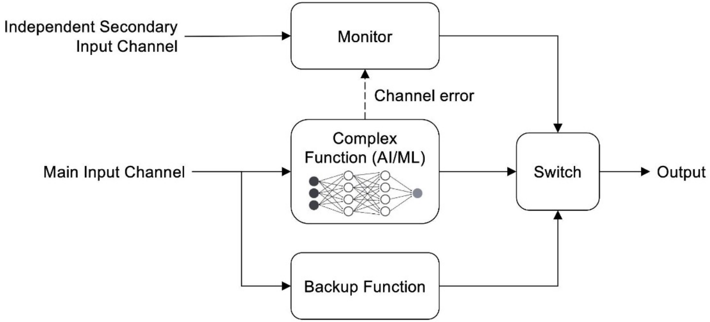


Combination of generic architectural patterns (2)

- Monitor now independently monitors the system state and status known to cause concern w.r.t. complex function performance, e.g., outside ODD

Challenges

- Defining the status that causes poor performance in the complex function with sufficient assurance



Combination of generic architectural patterns (3)

- Monitor now compares output of the complex function to a defined criteria
- Trying to take advantage of higher assurance conventional backup for better availability

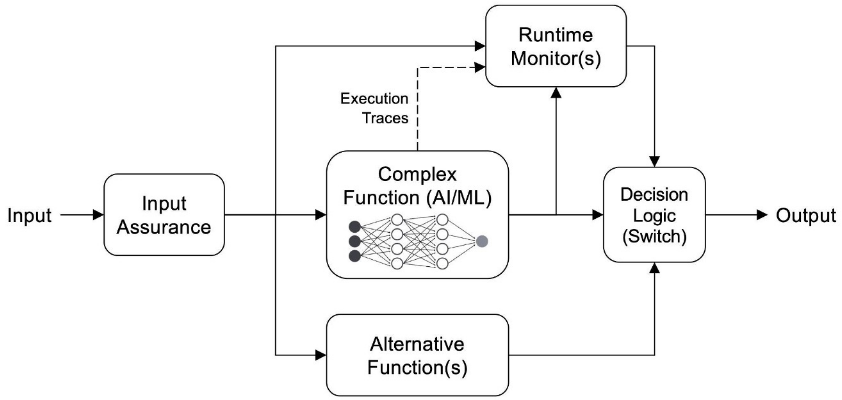
Challenges

- Complexity of defining the criteria checked by the monitor
- May include timing behaviours?

Architectural Patterns for AI/ML (1)

Runtime Assurance (RTA)

- Observe external/internal system state and state changes (including environment)
- Invoke alternative function when observer aka monitor signals violation
- Alternative function may or may not be a full functional equivalent
  - Safety-critical vs. Mission-critical
- Not a new pattern per-se
  - Auto-GCAS, RAIM, ECM, IVHM, FDIR
  - Increasingly being recommended for untrusted complex functions (ASTM F3269-21)
  - “Wrap” AI / ML-based function



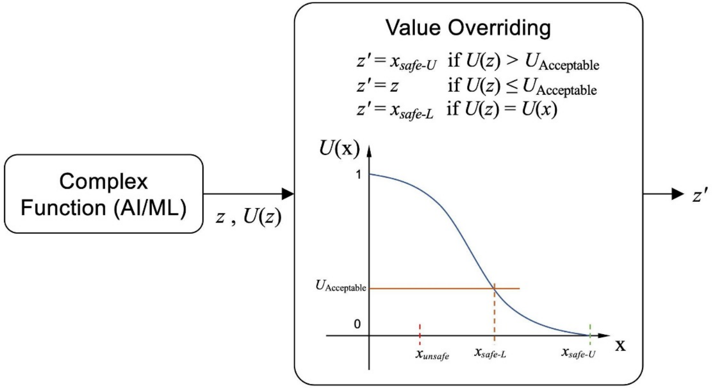
Challenges

- Choice of complex function boundary
- Monitor specification and development
- Assurance requirements allocation
- Configurations
- Decision logic specification complexity

Architectural Patterns for AI/ML (2)

Value Overriding

- Proposed in automotive domain
- Abstraction of 2(4) variants: uncertainty supervisor, safety margin selector, *adaptive* versions
- Replace value with *safe value*, predetermined uncertainty threshold



Challenges

- Admits safety margin reduction for performance gains in lower-risk operating situation
  - Violates fail-safe design principles of airworthiness regulations
- Safe value / reference uncertainty distributions must exist and be independently determined and validated

Challenges

- Uncertainty estimates produced as input must themselves be trusted to apply thresholding
  - Incorrect responses with high confidence
- Operating situations are correctly determined
  - This is itself the perception problem → circularity

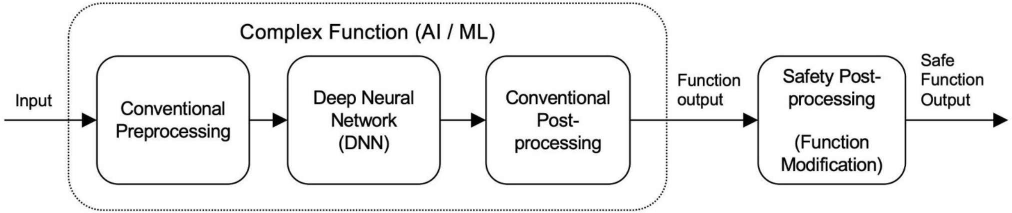
Architectural Patterns for AI/ML (3)

Function Modification

- Replace function output with *safe* output of *safety post-processing*
  - Could be seen as analogous in intent to Value Overriding
- Again, proposed in the automotive domain
  - For inaccurate localization
  - Scale bounding box by enlargement factor proved to always contain ground truth

Challenges

- Applies only to true positive detections
- Cannot correct false detections



Challenges

- Requires assurance that complex function is robust and behaves as expected for in-domain, in-distribution inputs
- Cannot be applied in single-channel configuration
  - Requires combination with other patterns and consolidated analysis of safety contribution, e.g., OOD detection monitor, self-checking pair, active monitor parallel design

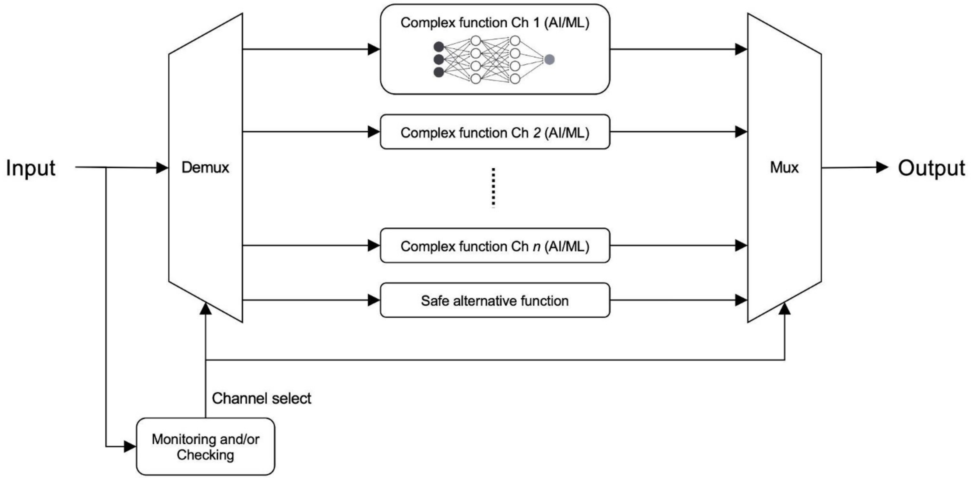
Architectural Patterns for AI/ML (4)

Input Partitioning and Selection

- Has been applied to Applied to ACAS-Xu
- Multiplexer (input) - demultiplexer (output)
  - Route partition of inputs to specific channel based on predefined conditions
- Two or more channels,
  - At least one safe alternative channel not including ML/AI

Challenges

- All channels must be available; loss of channel → loss of function for a specific partition of inputs
- High assurance of correctness of monitor or input-output selection logic
  - e.g., Select backup only in those regions where response of primary and backup are known to diverge
  - Inputs are correctly routed



Challenges

- Primary channels and safe backup channel are shown correct and consistent with high assurance on some common portion of the operating domain
- Safe backup must be fully functionally equivalent → it is itself a complex function
- No relief in assurance of primary despite safe backup

Concluding Remarks

- First work to examine architectural patterns from safety standpoint when integrating ML in aviation
    - Not a comprehensive study of all known patterns
- HW/SW architectural patterns analysed from safety standpoint by others
    - But does not consider ML/AI integration
- Functional safety patterns (for “autonomy”)
    - For automotive domain, considering SILs
    - SIL concept can be applied to aviation, but unclear how to verify for ML/AI
    - Not the same as DALs, as with aviation applications
- Existing architectural patterns from aviation appear to remain valid
    - Need re-assessment when integrating ML/AI from DAL standpoint
    - Adjustments/modifications may be needed
- New patterns need careful assessment
    - Stringency of DAL requirements for high-criticality suggests implementation may be costly and effort intensive
    - Some new patterns may not be suitable in their current form (e.g., value overriding)
- For some patterns, main benefit may only be performance improvement
    - Reconciling with safety remains a major challenge.

Future Work

When Integrating AI / ML

- More comprehensive analysis of other patterns used in aviation
- Alignment with others architectures
    - Integrated Modular Avionics (IMA)
    - Full Airborne Capability Environment (FACE)
    - Common Avionics Architecture System (CAAS)
- Creating
    - An architectural patterns catalogue
    - Architectural patterns assessment framework
        - For evaluating credibility, suitability, safety contribution
    - Assurance case / confidence case patterns catalogue to accompany architectural patterns