UNIVERSITY of York

ASSURING AUTONOMY INTERNATIONAL PROGRAMME

Towards a Framework for Safety Assurance of Autonomous Systems

John McDermid, Yan Jia, Ibrahim Habli
University of York, UK

Assuring Autonomy & AI

A Multi-disciplinary Challenge

- Technical
  - Gaining confidence in AS, e.g. especially artificial intelligence (AI) or machine learning (ML) in complex, open environments
- Ethics
  - For example, what decisions should an AS be allowed to make, and how do we avoid biases, etc.?
- Regulatory and legal
  - How do we control innovation, without stifling it, in an international context?
- Social
  - How do we ensure AS are net beneficial to society?

Overview

Technical Issues

- Motivating Examples
- Solution Elements
- Framework
- Conclusions

Overview

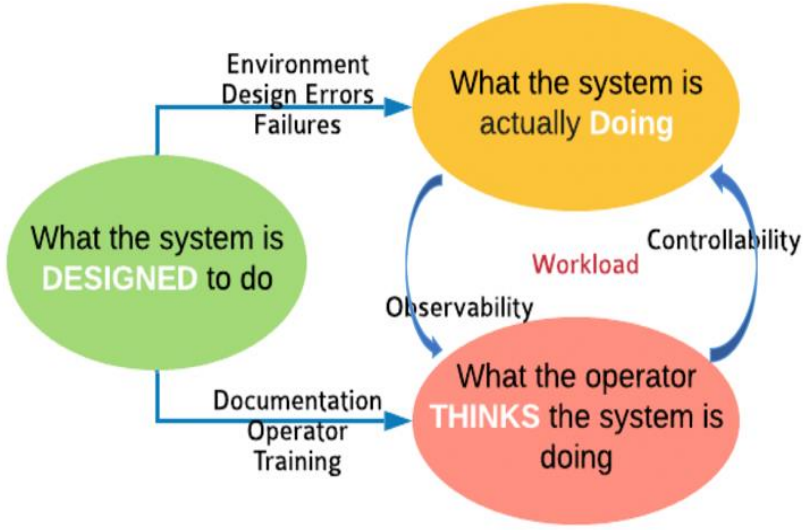
Technical Issues

- Motivating Examples
  - Watchkeeper Accidents
  - Quantitative Risk Assessment
  - AV Decision Making
  - AV Perception
- Solution Elements
- Framework
- Conclusions

Watchkeeper

Accidents and “gaps”

- Military “drone”
  - Classical safety process
  - Lost five in 15,000 hours
  - Far higher than predicted
- Gaps between
  - Model of system as designed
  - Actual behaviour
  - Operator model of behaviour
  - Don’t employ AI, but …



Quantitative Risk Analysis

Review by Rae et al (2014)

- Showed that QRA not accurate (NB Watchkeeper)
  - Stratified causes of inaccuracies
  - Produced a maturity model
    - Intent to 'fix' level 1 issues before moving to level 2
  - Not focused on systems employing AI
    - But many of the issues apply to AI, e.g. §2.3 and §2.4

2.3	Mismatch between the risk assessment and reality (discussed in Section 3.3.7)	
2.3a		Recommendations for action are inconsistent with assumptions in the risk assessment
2.3b		Risk assessment has been performed on an incorrect or misunderstood description of the system
2.3c		Invalid assumptions are made about the detectability of problems
2.3d		Invalid assumptions are made about the effectiveness of mitigations
2.3e		The required or designed behaviour of the system is assumed to be safe
2.4	Major inaccuracies in the analysis (discussed in Section 3.3.8)	
2.4a		Models are used outside their valid scope (including using models with little or no validity)
2.4b		Factors that significantly increase or decrease risk for specific groups, locations, or times are ignored (including effects of system ageing)
2.4c		Methods or models are applied incorrectly

Autonomous Vehicles

Inappropriate Decision-Making

Autonomous Vehicles

Inappropriate Perception

Overview

Technical Issues

- Motivating Examples
- Solution Elements
  - Safety-I and Safety-II
  - Desiderata for Machine Learning
  - Body of Knowledge
- Framework
- Conclusions

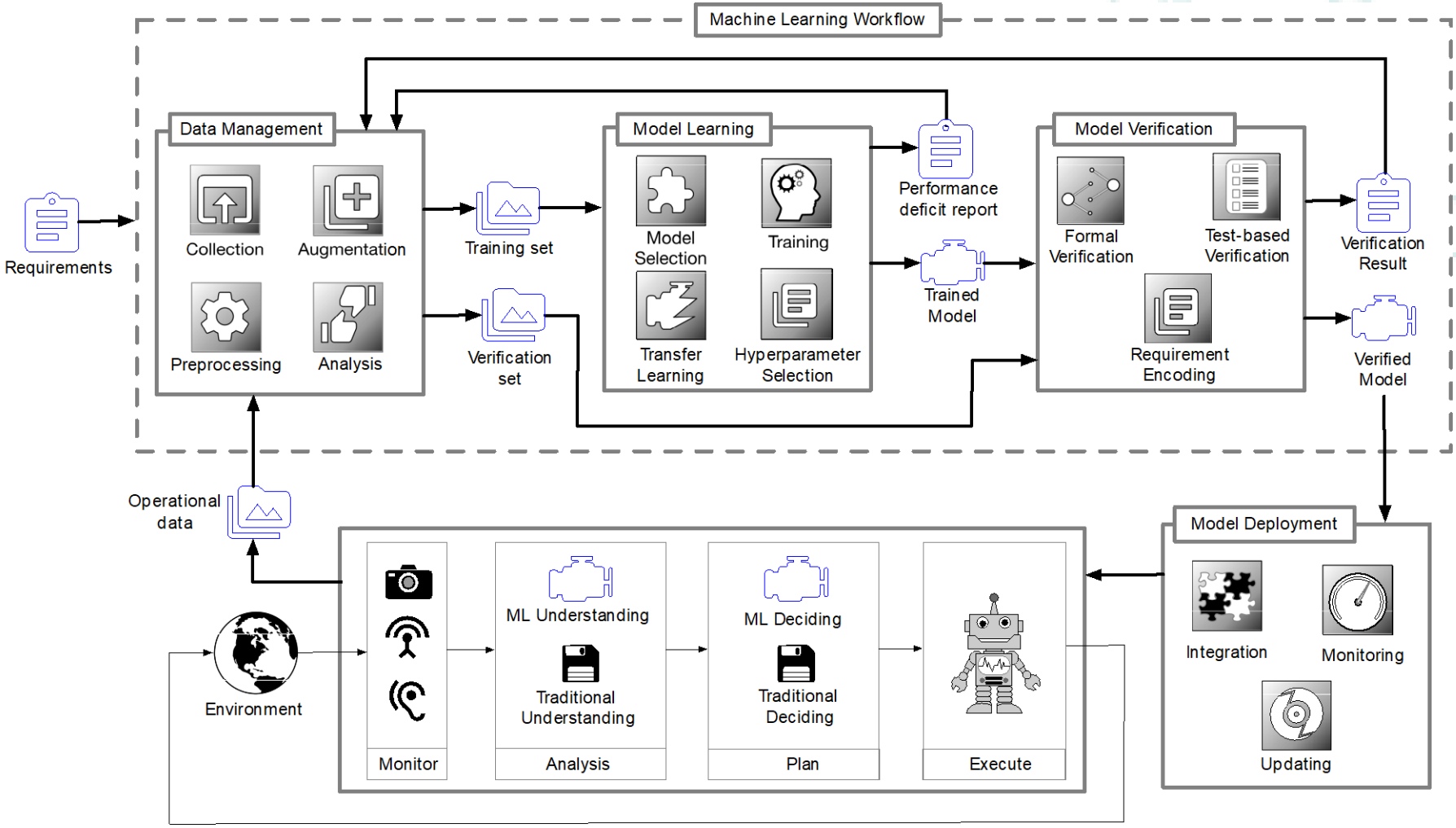
Safety-I and Safety-II

Philosophy due to Hollnagel

- Safety-I – focus on eliminating failures and errors
- Safety-II – focus on reinforcing ‘what goes right’
- Emphasizes the distinction
  - Work-as-imagined – how work is thought of either when it is being planned or when it occurs
  - Work-as-done – how work is actually carried out, where and when it happens
- Safety improvement
  - Reducing the gaps between work-as-imagined and work-as-done, recognising that neither is absolute

ML Life Cycle Model

Approach due to Ashmore et al



ML Desiderata

Approach due to Ashmore et al

Table 4. Open challenges for the assurance concerns associated with the Model Learning (ML) stage

ID	Open Challenge	Desideratum (Section)
ML01	Selecting measures which represent operational context	Performant (Section 5.4.1)
ML02	Multi-objective performance evaluation at run-time	
ML03	Using operational context to inform hyperparameter-tuning strategies	
ML04	Understanding the impact of hyperparameters on model performance	
ML05	Decoupling the effects of perturbations in the input space	Robust (Section 5.4.2)
ML06	Inferring contextual robustness from evaluation metrics	
ML07	Identifying similarity in operational contexts	Reusable (Section 5.4.3)
ML08	Ensuring existing models are free from faults	
ML09	Global methods for interpretability in complex models	Interpretable (Section 5.4.4)
ML10	Inferring global model properties from local cases	

support reuse [2]					☆
Transfer Learning [173]	✔	✓	✔	★	☆
Use model zoos [58]	✔	✓	✔	★	
Post-hoc interpretability methods [3, 93, 105]		✔			★

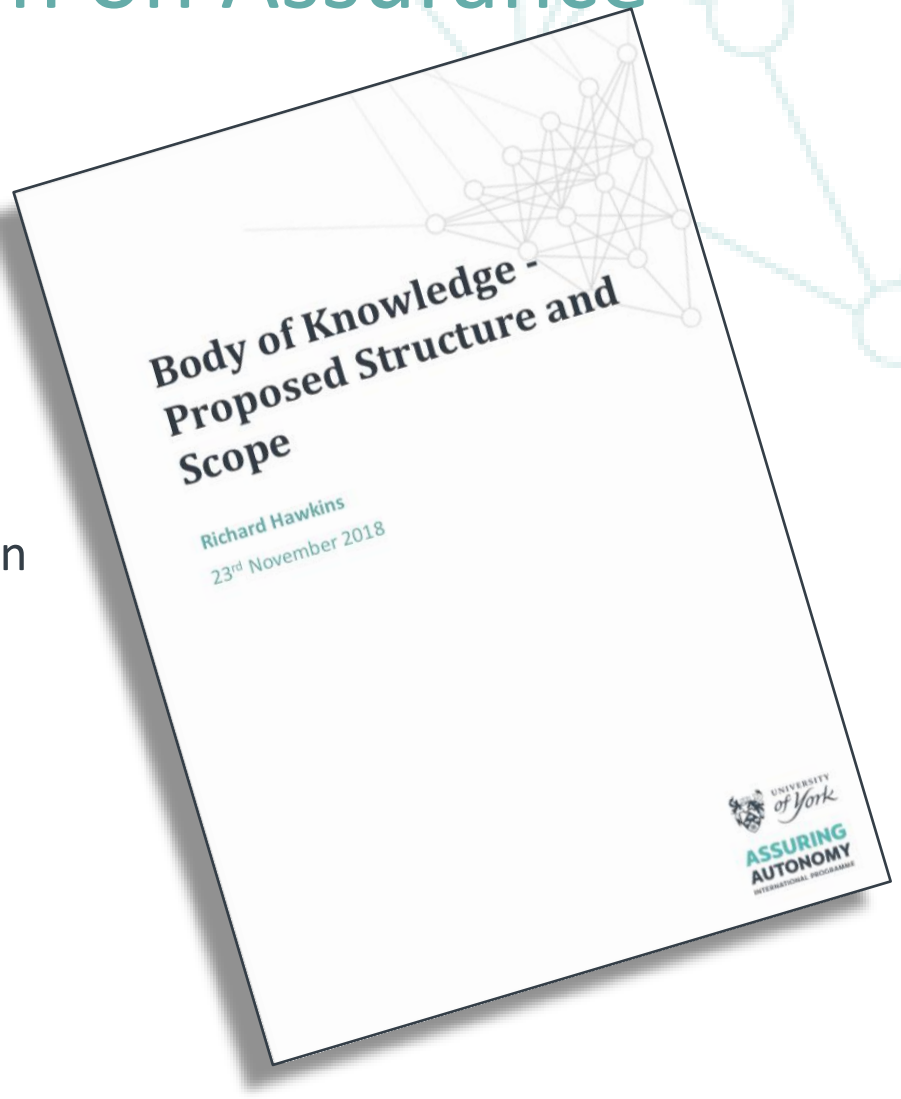
†✔ = activity that the method is typically used in; ✓ = activity that may use the method

‡★ = desideratum supported by the method; ☆ = desideratum partly supported by the method

Body of Knowledge

Source of Information on Assurance

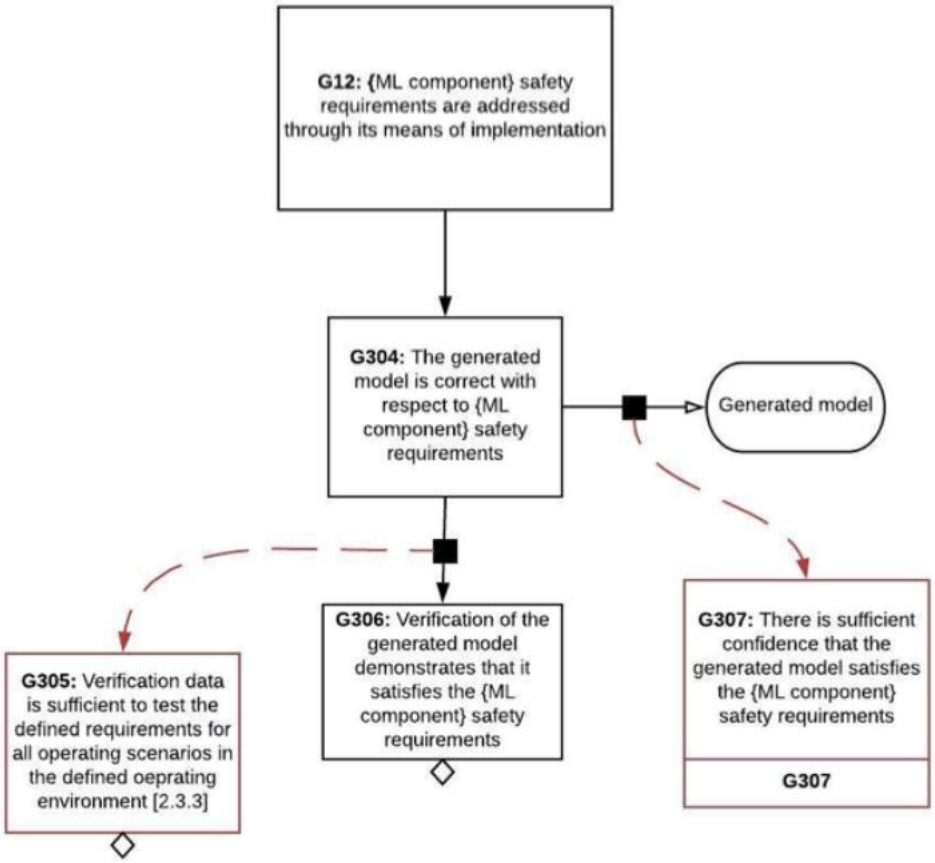
- Development of the BoK
  - Based on models of ML and system (MAPE/SUDA)
  - Each element supported by
    - Objectives
    - Approaches to demonstration
    - Contextual information
  - Initial on-line version
    - Will evolve, incorporating results of programme and other work



Body of Knowledge

Proposed Safety Case Pattern

- Safety case needs to address characteristics of ML
  - Fragment of pattern shown in GSN
  - Confidence arguments key to addressing ML
- Not yet clear if can be general or need to be domain specific



Overview

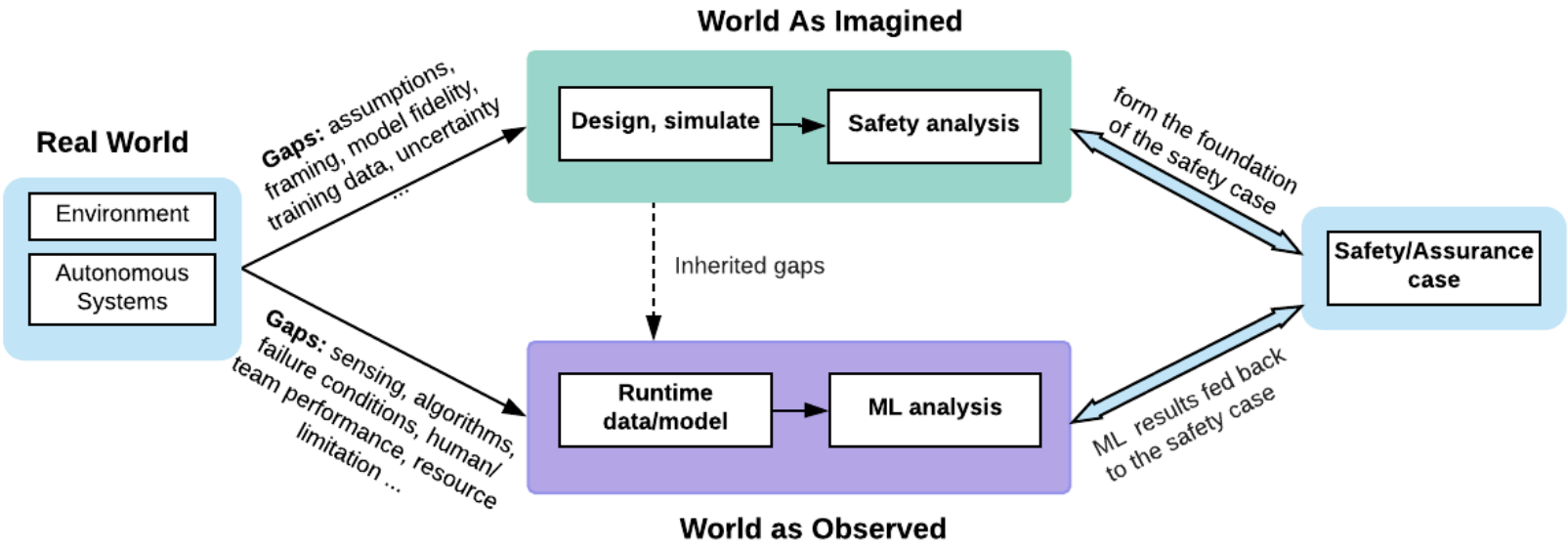
Technical Issues

- Motivating Examples
- Solution Elements
- Framework
  - The framework and work-as-observed
  - ML as part of the solution
  - Safety Case
  - Regulation
  - Illustration
- Conclusions

Framework

General Structure

- Extends and actualizes Hollnagel's philosophy
  - World-as-observed what we can understand of real-world through analysis of (system) data
  - Safety case enables reduction of gaps …



ML as A Partial Solution

Ability to Analyse and Compare

- AS are (usually) data rich
  - ML allows patterns of real-world behavior to be identified and assessed (dependent on data collection)
- ML can show
  - Behaviour at variance to what was imagined
- Feedback
  - Potential improvements to system design, operational procedures, data collection
  - Enables reduction of the gap between the world-as-imagined and the real-world

Safety Case

Informed by Imagination & Observation

- Initial
  - Based on work-as-imagined – fairly classical, but needs to cover all the gaps, including those due to ML (show how they are managed)
- Evolving
  - Updated based on work-as-observed – information to support feedback into practice (design, procedures …)
- Ultimately dynamic
  - Analysis in work-as-observed close to real-time, perhaps ultimately allowing risk-aware safety management

Regulation

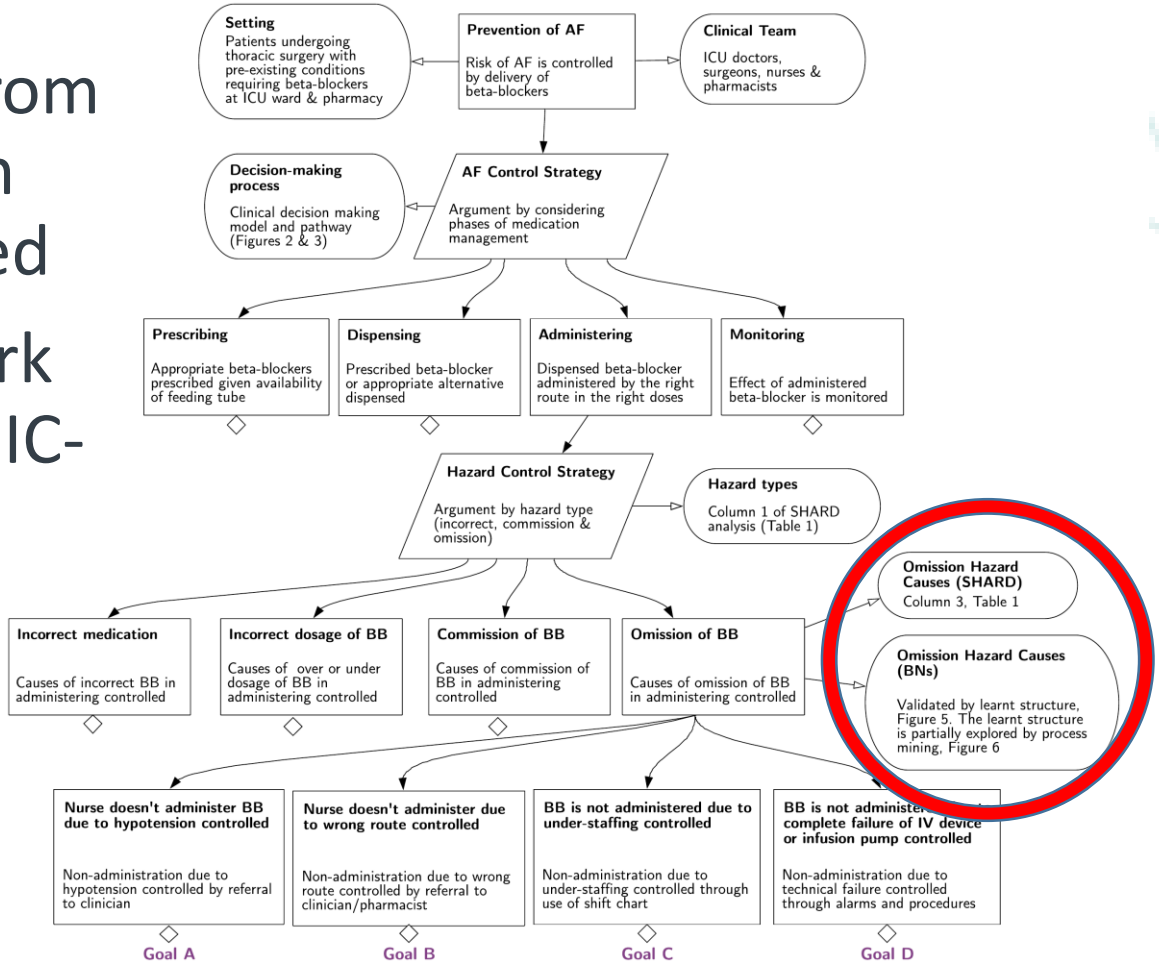
Regulatory Processes need Revision

- Current regulatory processes
  - Effectively assume analysis pre-operation is ‘for life’
  - Revise (only) in the event of major design change or accident
- Current processes unsustainable for AI/AS
  - Analysis of QRA and WK show already challenged
  - Behaviour changes in operation make processes untenable
- Revised Processes
  - Much more incremental
  - Initial approval based on work-as-imagined
  - Need to update based on work-as-observed

Illustration

Medication Safety after Thoracic Surgery

- Hazard causes from safety analysis in work-as-imagined
- Bayesian Network analysis on MIMIC-III data
  - Showed a gap (misalignment)
  - Refined safety case shown



Overview

Technical Issues

- Motivating Examples
- Solution Elements
- Framework
- Conclusions

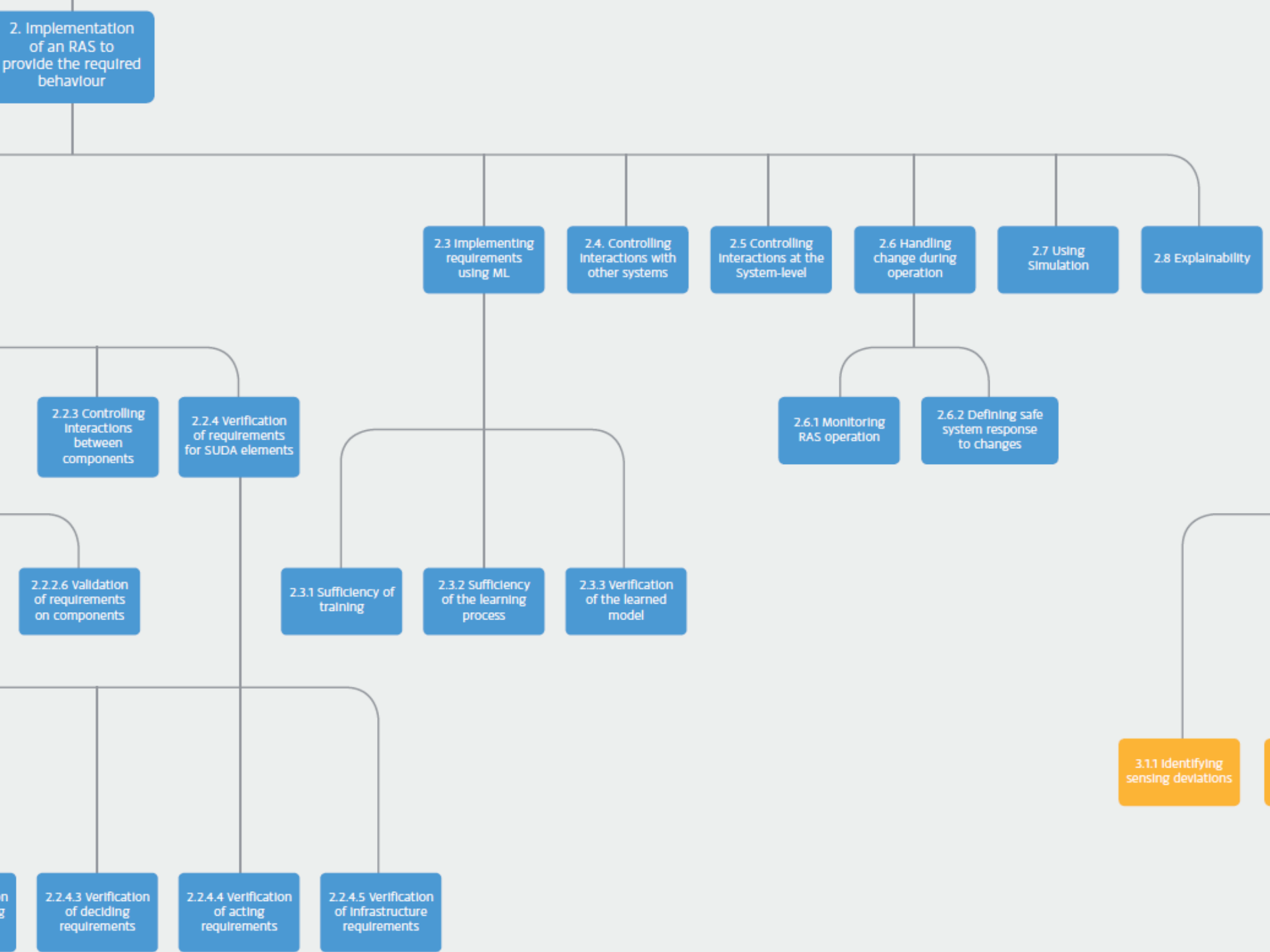
Conclusions

Mind the Gap(s)

- AI and AS can't be assessed effectively using current safety and assurance processes
  - In the framework, initial analysis is fairly conventional, but the safety case needs to address the 'gaps'
  - The proposed framework includes mechanisms for identifying and helping to reduce 'gaps'
  - Sees ML as part of the solution, as well as a 'problem'
- Concept of 'gaps' also applies in ethical, regulatory/ legal (and social?) contexts
  - A basis for an AI (autonomy) safety landscape?

ASSURING AUTONOMY
INTERNATIONAL PROGRAMME

Funded by
Braford Teaching Hospitals NHS Foundation Trust


2. Implementation of an RAS to provide the required behaviour
2.3 Implementing requirements using ML
2.4. Controlling interactions with other systems
2.5 Controlling interactions at the System-level
2.6 Handling change during operation
2.7 Using Simulation
2.8 Explainability
2.2.3 Controlling interactions between components
2.2.4 Verification of requirements for SUDA elements
2.6.1 Monitoring RAS operation
2.6.2 Defining safe system response to changes
2.2.2.6 Validation of requirements on components
2.3.1 Sufficiency of training
2.3.2 Sufficiency of the learning process
2.3.3 Verification of the learned model
3.1.1 Identifying sensing deviations
2.2.4.3 Verification of deciding requirements
2.2.4.4 Verification of acting requirements
2.2.4.5 Verification of infrastructure requirements