Developing a Taxonomy of Faults for the Integration of Autonomous Components

Chris Allsopp

02/07/2020

SYSTEMS AND ENGINEERING TECHNOLOGY

Introduction

- Personal introduction
- Outline the problem and context
- Introduce the research project and our approach
- Explain our proposed fault taxonomy and examples
- Describe next steps
- Questions and feedback
  - After the event please email: c.allsopp@fnc.co.uk

2

Chris holds Bachelor and Masters degrees in artificial intelligence but spent the first decade of his career developing safety critical systems and software across a range of industries. Over the last 2 years Chris has focussed on autonomous systems. He leads a multidisciplinary team developing autonomy related technology and autonomous systems primarily for the defence market and supports a number of projects and conferences tackling the issues of assuring autonomy.

Disclaimer

- This is an overview of UK MOD sponsored research and is released for informational purposes only. The contents of this presentation should not be interpreted as representing the views of the UK MOD, nor should it be assumed that they reflect any current or future UK MOD policy. The information contained in this presentation cannot supersede any statutory or contractual requirements or liabilities and is offered without prejudice or commitment.
- The project is ongoing and the taxonomy is a work in progress presented to stimulate discussion and feedback

Problem Outline

- Autonomous systems and related AI technologies can deliver enhanced system capabilities
- Current V&V techniques, assurance approaches and safety standards have not kept pace with technological change
- Adoption of autonomous technologies is likely to be phased
  - Incremental upgrades of existing conventional systems
  - Step changes in capability from systems designed to be autonomous from day one
- Exposes key integration challenges
  - Need to integrate autonomous components into existing, conventional systems
  - Understand how to integrate autonomous systems with other autonomous systems

4

SYSTEMS AND ENGINEERING TECHNOLOGY

- The interest in autonomous systems and enabling technologies is due to the potential they offer beyond the capabilities of conventional systems. I'm sure the community at this event are working on a broad set of problem domains in defence, automotive, agriculture and more.
- The technology is progressing and evolving very quickly due to the level of public and private investment in the field. As is typically the case, assurance techniques and standards haven't been able to keep pace with the technology development. That will be a barrier to bringing autonomous systems into operation in the real world, hence the need for groups like the IEEE TC, the Safety Critical Systems Club RAS Working Group and a number of others.
- In the defence domain it's likely that the adoption of autonomous technologies is going to be phased. Autonomous components will be integrated into conventional systems to deliver capability enhancements and also help to understand the broader issues with bringing these technologies into an operational environment.
- Longer term there'll be systems developed that are designed to be autonomous from the ground up.
- Both of those scenarios expose integration challenges that are likely to have commonality but also differences. As a community we need to understand how to integrate autonomy into existing conventional systems and how to integrate autonomous components and systems with each other.

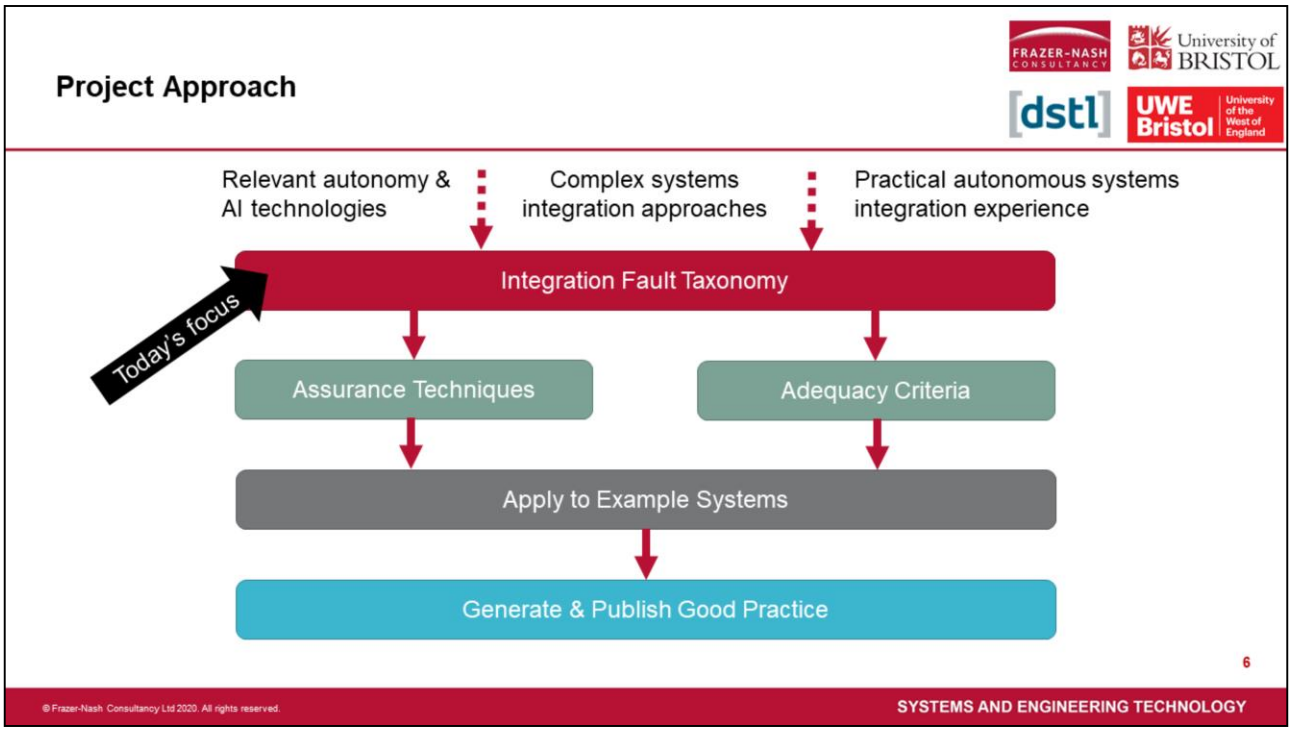
Project Background

- Aim: Develop guidance and approaches that will contribute to a suite of good practice for the assurance of autonomous component **integration**
- Part of a broader Dstl research programme into the assurance of autonomous systems
- Collaboration between industry and academia:
  - Frazer-Nash Consultancy
  - Professor Kerstin Eder, Head of the Trustworthy Systems Lab at the University of Bristol
  - Professor Jim Smith, Head of the AI research group at the University of the West of England
- Interaction with research community a key objective
  - Feedback and discussion welcome and encouraged – email c.allsopp@fnc.co.uk

5

SYSTEMS AND ENGINEERING TECHNOLOGY

- The aim of this research project is to develop guidance and approaches to support the effective and assured integration of autonomous components. This will contribute to the creation of good practice and ideally development standards in time.
- The focus is very much on integration, we're not aiming to solve the whole autonomy assurance challenge. Although it would be nice if we could!
- This work feeds into a larger, wide ranging autonomous systems assurance research programme that Dstl are running on behalf of the UK government defence enterprise.
- We're running the project collaboratively to bring together diverse perspectives from industry and academia. From within Frazer-Nash we have both autonomy technology and safety focussed input. Professor Kerstin Eder, Head of the Trustworthy Systems Lab at the University of Bristol brings deep knowledge of V&V techniques and their applicability to autonomy-related technologies. Professor Jim Smith, Head of the AI research group at the University of the West of England has an encyclopaedic knowledge of the broad set of AI techniques applicable to this domain and their relative strengths and weaknesses.
- Interaction with the research community is a key objective made more difficult by the Covid-19 situation. I'd like to emphasise that we really would welcome and appreciate discussion and feedback during this event or afterwards. My email address is listed throughout the slides so please do get in touch.



- Our approach to the project has been to bring the group's experience of autonomy and AI technologies, complex systems integration and autonomous systems integration together to form a set of integration faults. We've then attempted to categorise those faults into a fault taxonomy.
- The aim is to be able to understand the nature of those faults and then be able to link them to assurance techniques that can prevent, detect and mitigate the faults. We'll also consider the adequacy of those techniques to understand where diverse complementary approaches may be appropriate.
- To make sure the work isn't too theoretical, and to validate our approaches, we'll apply some of them to example systems. The group is working on a range of physical systems that will provide useful feedback on the efficacy and practicality of our suggested techniques.
- We aim to publish our outputs through a range of channels such as conferences and research community groups.
- The focus of today's presentation is the fault taxonomy.

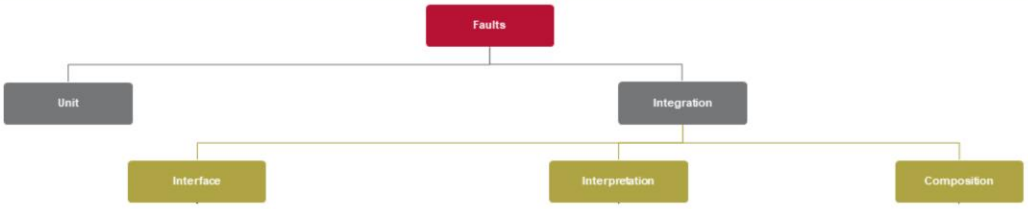
Proposed Fault Taxonomy

7

SYSTEMS AND ENGINEERING TECHNOLOGY

- To reinforce the point that we're focused on integration faults we've created two categories at the top level. One for faults within the unit or component and one for faults that occur during integration.

Proposed Fault Taxonomy

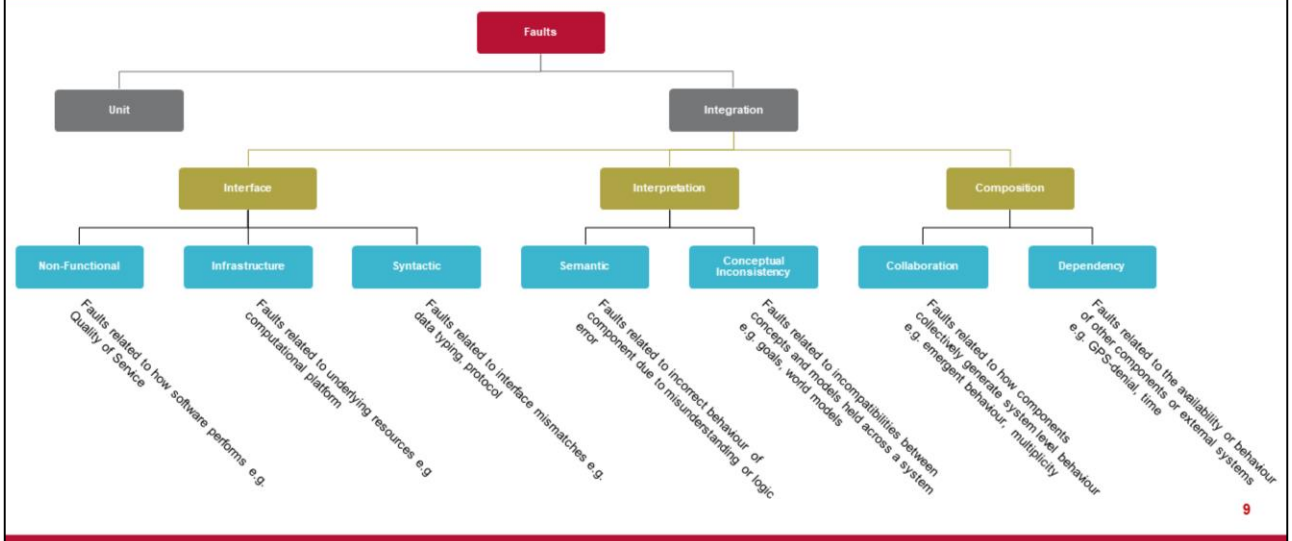


8

SYSTEMS AND ENGINEERING TECHNOLOGY

- The next level of the taxonomy is split into three categories:
  - Interface faults are classic integration problems that you see when components don't communicate with each other effectively.
  - Interpretation faults are a little more abstract and relate to components having a different understanding of how they are supposed to behave.
  - Composition faults are associated with the way components operate together to deliver the system's functions.

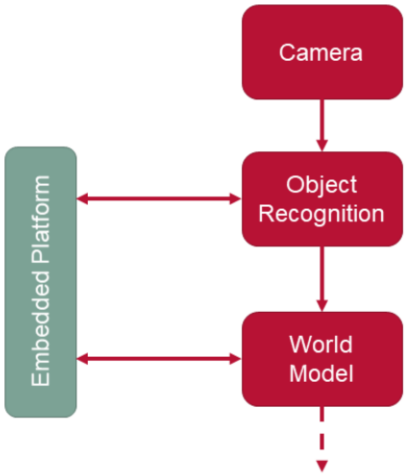
Proposed Fault Taxonomy



SYSTEMS AND ENGINEERING TECHNOLOGY

- At what is currently the lowest, most concrete level of the taxonomy we have seven categories.
- See definitions of the categories on the slide.
- The categories are illustrated with examples on the following slides.

Example 1 – Vision System



Fault Type

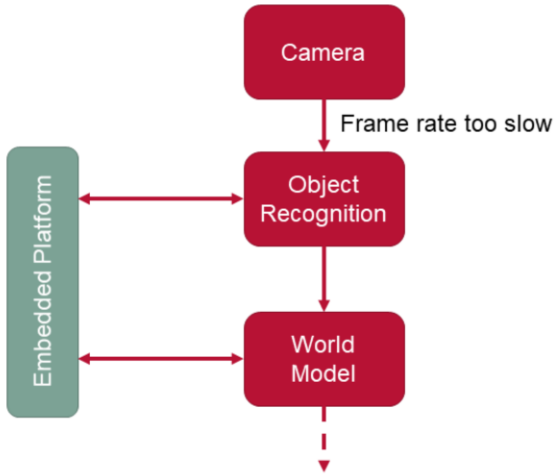
- Non-functional
- Infrastructure
- Syntactic
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

10

SYSTEMS AND ENGINEERING TECHNOLOGY

- The first example is a vision system consisting of a camera, an object recognition algorithm and a world model that feeds into a wider system. Both of the object recognition and world model components run on an embedded computing platform.

Example 1 – Vision System



Fault Type

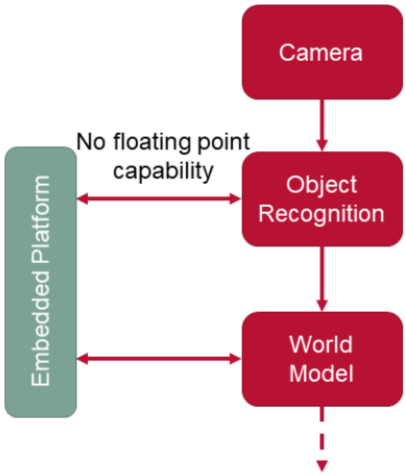
- Non-functional
- Infrastructure
- Syntactic
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

11

SYSTEMS AND ENGINEERING TECHNOLOGY

- The camera could be providing video at 15fps when the object recognition component is expecting 30fps. We'd consider this a non-functional fault as it relates to a Quality of Service property of the interface.

Example 1 – Vision System



Fault Type

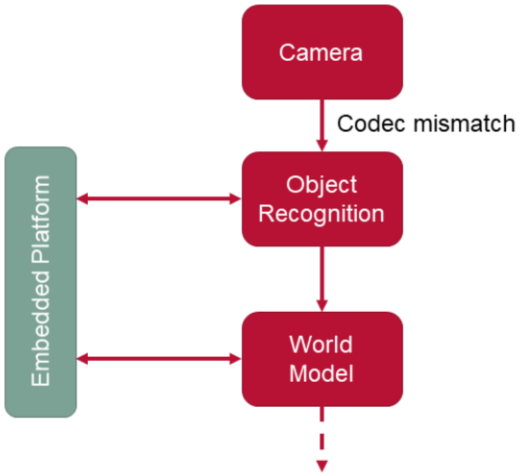
- Non-functional
- Infrastructure
- Syntactic
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

12

SYSTEMS AND ENGINEERING TECHNOLOGY

- Imagine that the object recognition component is a deep learning model that has been trained on a high performance computing platform and then deployed to the embedded platform for operation.
- The nature of the embedded platform may not have been known by the object recognition development team so assumptions have been made and constraints not applied to the model development.
- When deployed, the model can't run because the embedded platform does not have a floating point unit or suitable emulation. We'd consider this to be an infrastructure fault.

Example 1 – Vision System



Fault Type

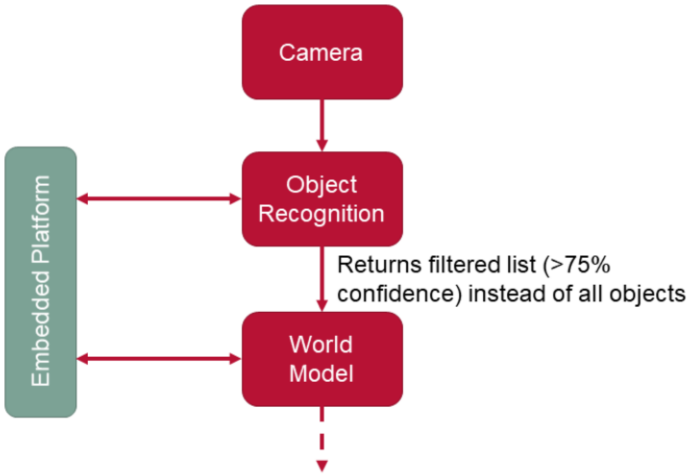
- Non-functional
- Infrastructure
- Syntactic
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

13

SYSTEMS AND ENGINEERING TECHNOLOGY

- A more fundamental integration error between the camera and the object recognition component would be mismatch in the video codecs being used by the components. For example the camera delivers H.264 and the object recognition component expects MPEG-4. This is a syntactic fault.

Example 1 – Vision System



Fault Type

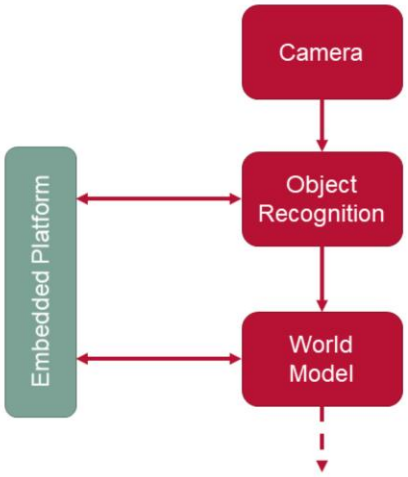
- Non-functional
- Infrastructure
- Syntactic
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

14

SYSTEMS AND ENGINEERING TECHNOLOGY

- The world model may be expecting to receive all objects recognised but instead only receives a list of objects detected with a confidence of >75%.
- This is a misunderstanding of the logic between components and the contents of the information being communicated so would be a semantic fault.

Example 1 – Vision System



OR: Tank
WM: None? or Armoured Vehicle?

OR: None?
WM: Tank

Fault Type

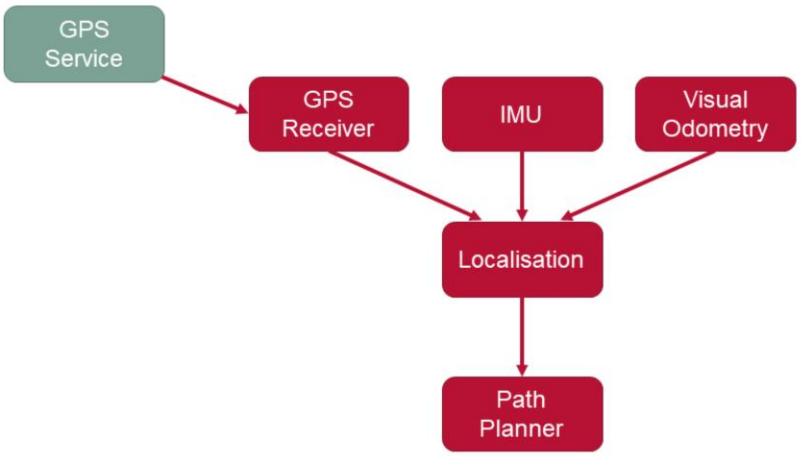
- Non-functional
- Infrastructure
- Syntactic
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

15

SYSTEMS AND ENGINEERING TECHNOLOGY

- Moving up in abstraction, consider a case where an existing agriculture system is being modified for a military use case. Given the different environment the system will be subjected to, you could imagine integrating an object recognition component trained specifically for the new military environment.
- This could result in faults arising from conceptual inconsistencies between components as illustrated here. The military specific object recognition component is likely to have a concept of a tank that aligns with the top image while the world model equates a tank to an agricultural storage tank which has very different properties and interest for the rest of the system. Object classes may also be present in one component but not the other.

Example 2 - Localisation



Fault Type

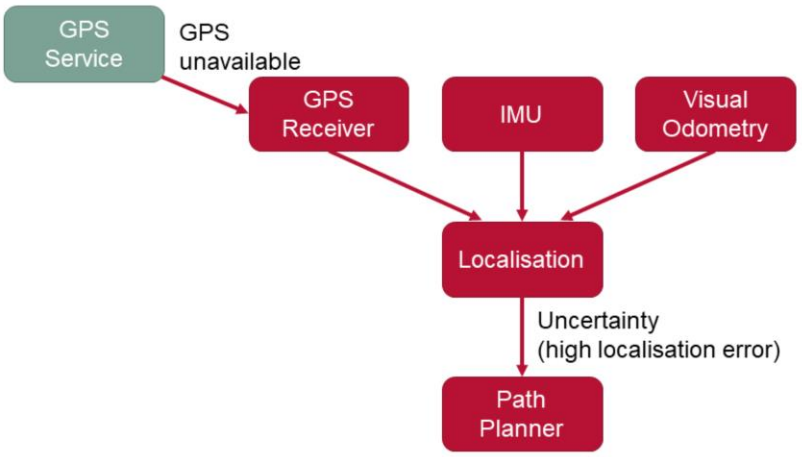
- Non-functional
- Infrastructure
- Syntactic
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

16

SYSTEMS AND ENGINEERING TECHNOLOGY

- Here we have a different example system consisting of three sensor inputs into a localisation component that provides current system location to a path planner.

Example 2 - Localisation



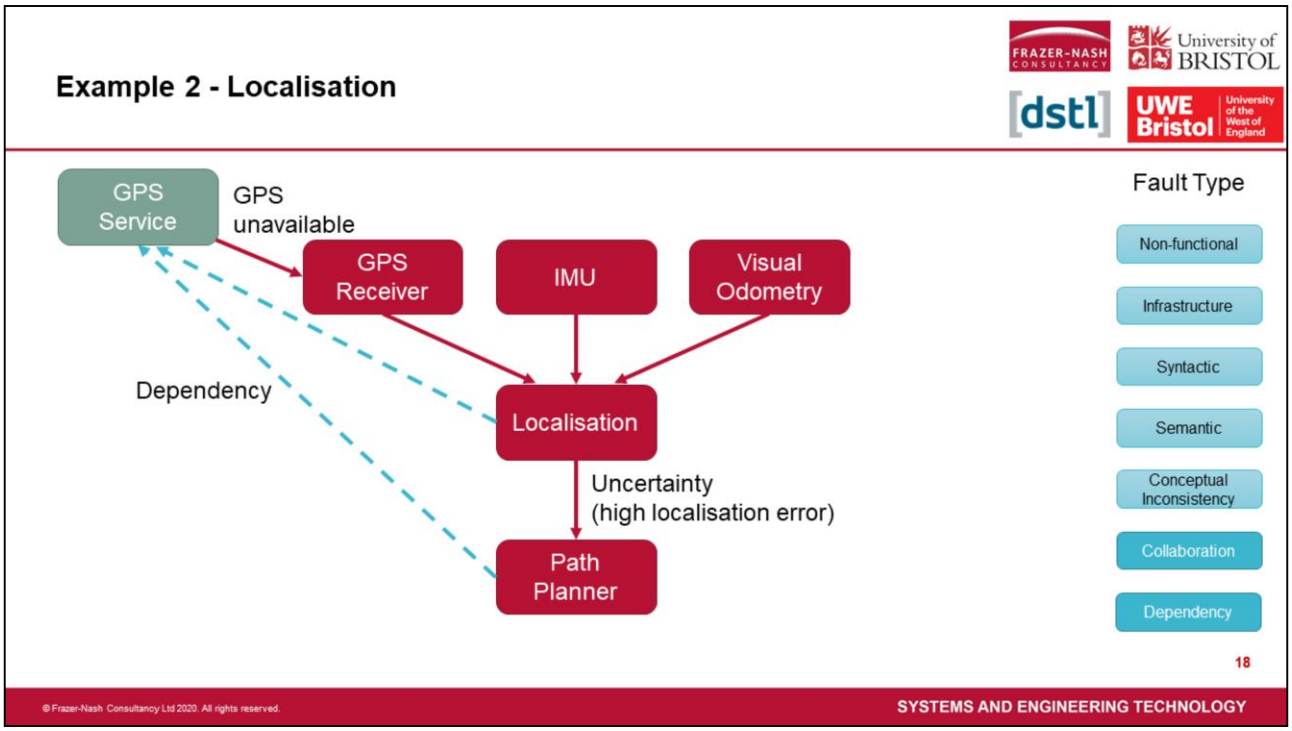
Fault Type

- Non-functional
- Infrastructure
- Syntactic
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

17

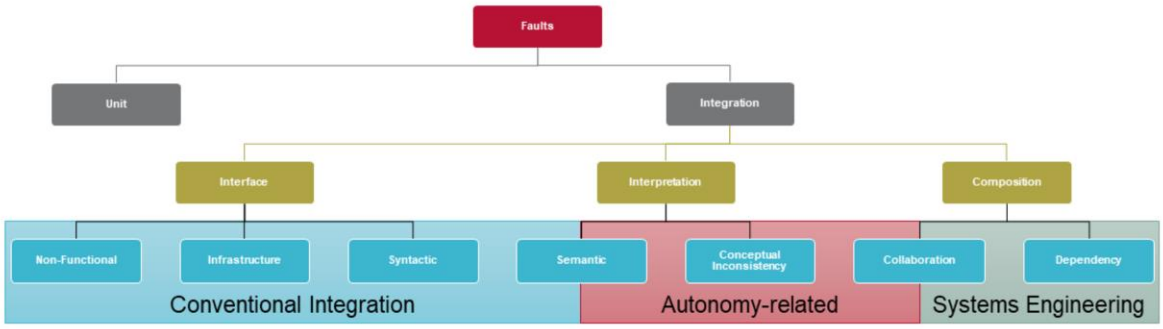
SYSTEMS AND ENGINEERING TECHNOLOGY

- In this example the GPS service is unavailable which could happen for a number of reasons such as being in a dense urban environment, GPS jamming or equipment failure.
- The unavailability of GPS causes an increase in localisation error and uncertainty due to the reliance on noisier measurement techniques, causing a failure of the path planning component.
- This is a collaboration fault that arises due to the interaction between components and assumptions regarding the level of uncertainty that components are able to cope with.



- It could also be considered a dependency fault because, in the context of this integrated system, the localisation and path planner components are dependent on the availability and correct function of the GPS service.
- The implicit dependency of the path planner on the GPS service may not have been clear at development time as an assumption had been made on the level of uncertainty in the location being provided to the component. This assumption has been violated in this case.

Proposed Fault Taxonomy – How is Autonomy Different?



Provides good coverage of faults in scenarios where environments and systems are relatively **static**

19

SYSTEMS AND ENGINEERING TECHNOLOGY

- Taking a step back and looking at the taxonomy as a whole we were interested in understanding what the key areas of concern were for autonomous systems. We don't want to spend time recreating research into the nature of conventional integration issues.
- Our discussion highlighted that the faults associated with lower levels of abstraction, such as syntactic faults, were not significantly different in the autonomous systems domain.
- Likewise, understanding dependencies and some aspects of collaboration are in the classical systems engineering domain.
- The integration challenge is increased for autonomy where the abstraction is higher and we face conceptual challenges, emergent behavior, and the lack of transparency of some technologies magnifies the risks of semantic faults.
- This thinking holds for systems or environments which are relatively static and change is limited.

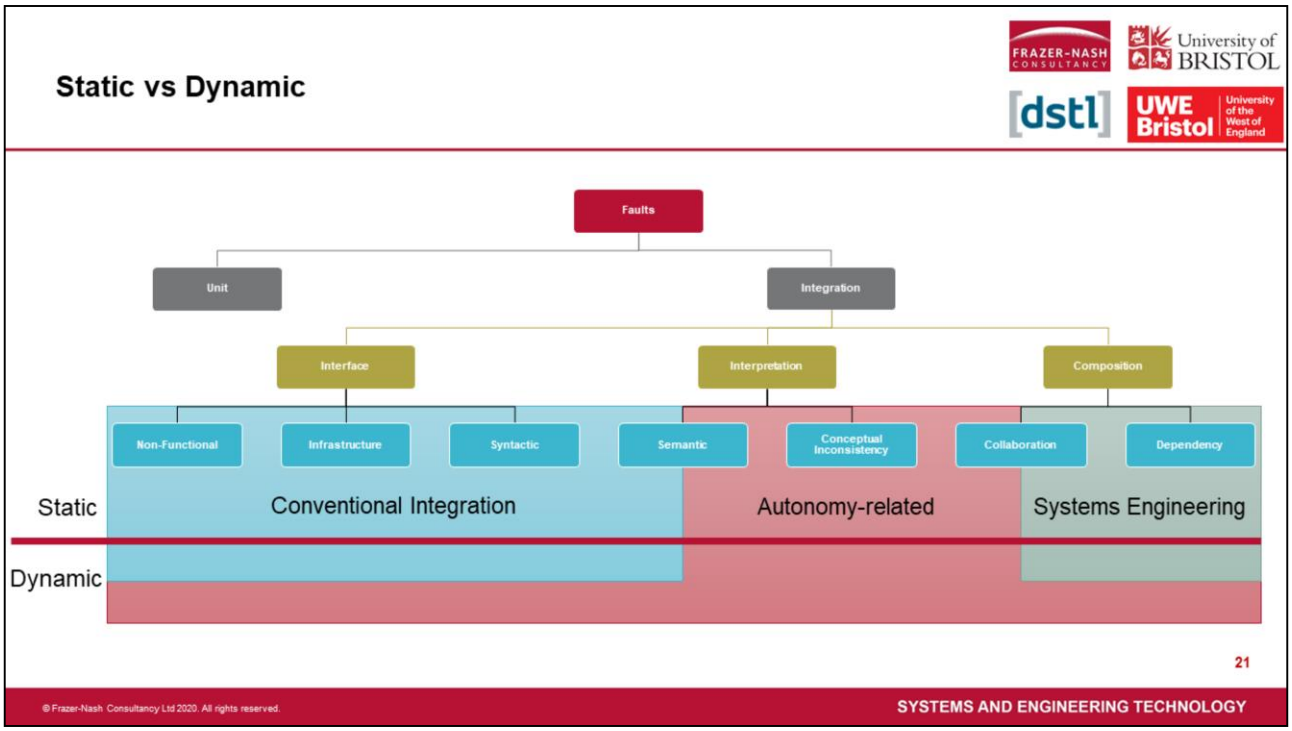
Dynamic Systems & Environments

- A key potential benefit of autonomous systems is the ability to operate in changing environments
- Most environments change in some way
  - Environment in the traditional sense e.g. weather patterns, seasonality
  - Entities within the environment change in number or form e.g. vehicle designs, fashion
  - Data environment that an algorithm operates in
- Systems can also change over time
  - Adaptation through online learning or evolution
  - Changing performance of system elements e.g. sensors, actuators, electronics
- Integration is implicitly occurring during system operation
  - How does this affect the types of faults observed?

20

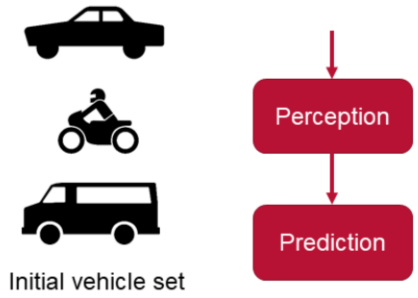
SYSTEMS AND ENGINEERING TECHNOLOGY

- Applicability to static systems or environments is a significant limitation as we know the real potential of autonomous systems lies in their ability to operate in changing situations and react appropriately to the unexpected.
- The reality is that most environments change in some way whether this is clear and obvious to us as humans or much more subtle and long term. The environment in a traditional sense certainly offers dynamic challenges with weather conditions and patterns, seasonality and other effects. Entities sharing the environment with an autonomous system can also change over time with examples like evolving vehicle designs and the clothes people wear presenting a challenge to self-driving car perception stacks. We should also consider that changes in the data environment of an algorithm can significantly impact performance, such as recommender systems where large number of users click through to unrelated or harmful items.
- Systems themselves can change over time, either by design or just through operation. The autonomous systems community is very interested in adaptive systems that use online learning or evolution to improve performance over time. More subtly, physically embodied systems will see changes in sensor, actuator or electronic characteristics over time that will affect behaviour and performance.
- As the systems and/or the environments change over time, integration is implicitly occurring during system operation. How does this affect the types of faults observed?



- This is a question that we're still exploring so I'll share some of the thoughts we have discussed so far.
- Fault categories that in a static context are largely unchanged from conventional systems engineering and integration may take on renewed relevance, as what was assumed at design time may no longer hold during operation. This suggests that assurance techniques cannot be confined to the development process and run time approaches need to be considered.
- Another idea is that the variation in environment or system exposes faults that still fit into the set of categories we are proposing. The environmental or system changes act as a generator or a trigger of integration faults. Again, this suggests a need for runtime techniques.
- The third idea is that the system or environmental variation leads to new kinds of faults that require additional categories in our taxonomy to capture them.
- I'll share some brief examples of these challenges in the following slides.
- We'd be particularly interested in opinions and discussion in this area as we are aiming to develop a fault taxonomy that is both useful to, and usable by, the autonomous systems community.

Example 3 - Environmental Variation



Fault Type

- Non-functional
- Infrastructure
- Syntactic
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

22

SYSTEMS AND ENGINEERING TECHNOLOGY

- Consider a system that perceives the world and predicts the future movement of vehicles. It has been developed to operate in an environment with cars, motorcycles and vans.

Example 3 - Environmental Variation

Environmental variation

Initial vehicle set

Perception

Prediction

Fault Type

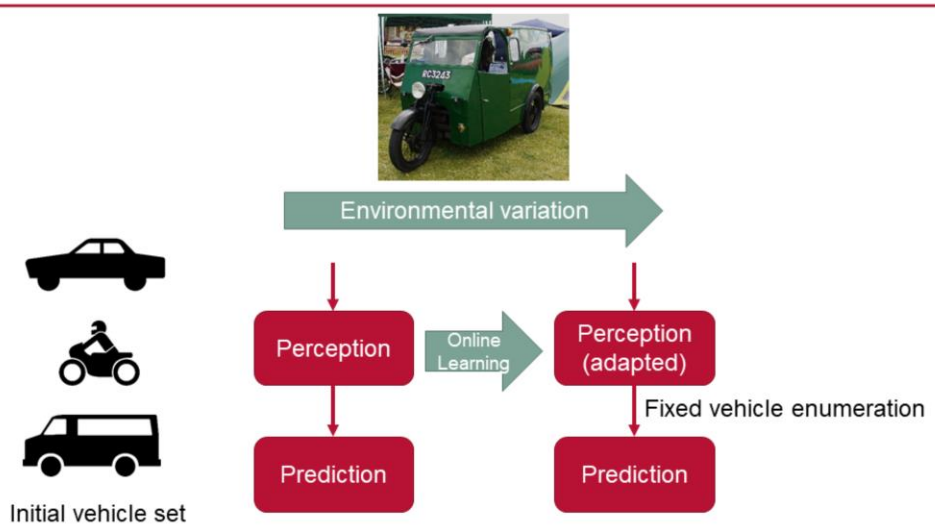
- Non-functional
- Infrastructure
- Syntactic
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

23

SYSTEMS AND ENGINEERING TECHNOLOGY

- Over time the nature of vehicles in the environment changes. The image shows an example of a hybrid between a motorcycle and a van, something that was not in the environment during development.

Example 3 - Environmental Variation



Fault Type

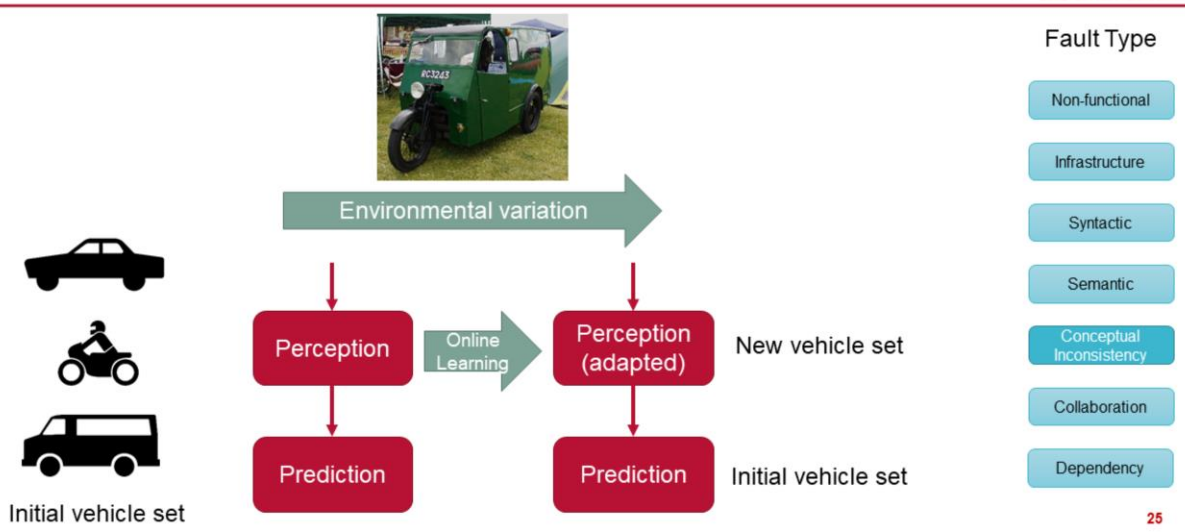
- Non-functional
- Infrastructure
- **Syntactic**
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

24

SYSTEMS AND ENGINEERING TECHNOLOGY

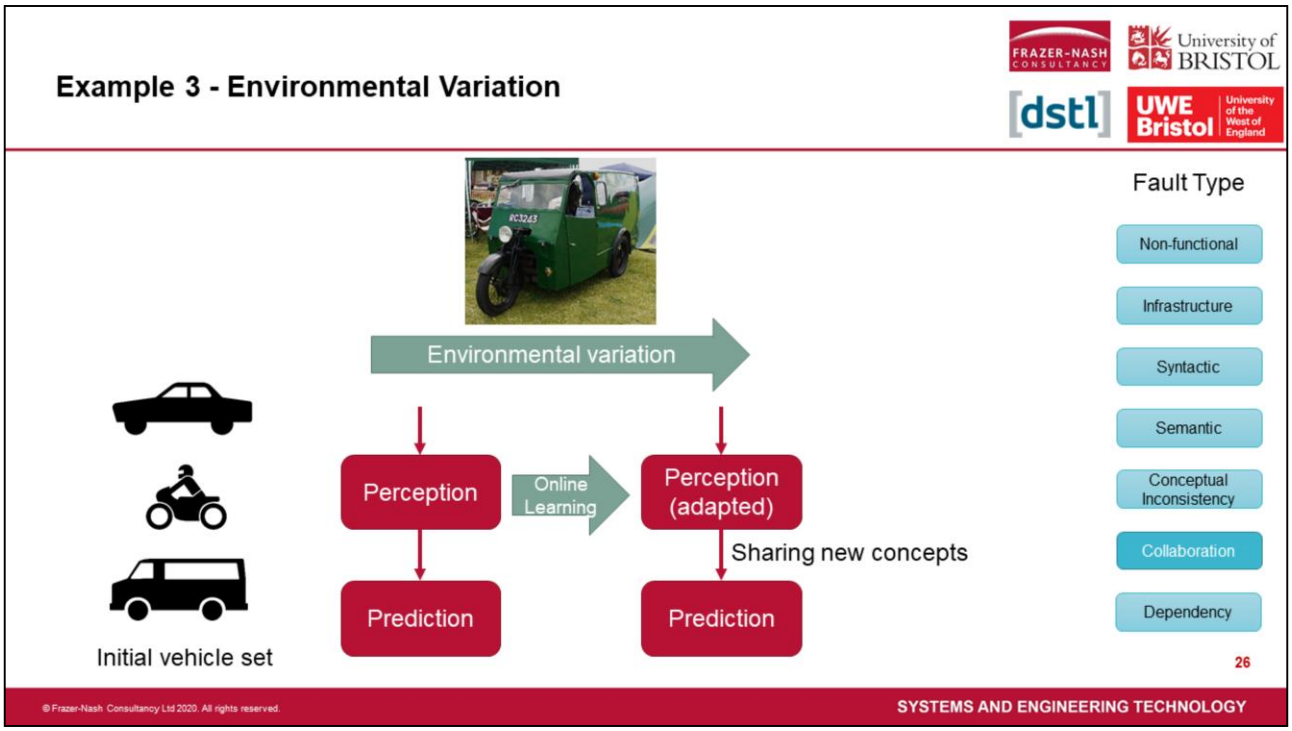
- In this example the system designers did not understand the impact of new classes being detected by the perception component on the prediction component. The interface definition does not support the communication of new concepts due to the use of fixed enumerated vehicle types. This would lead to a syntactic fault that could've been foreseen at design time.

Example 3 - Environmental Variation



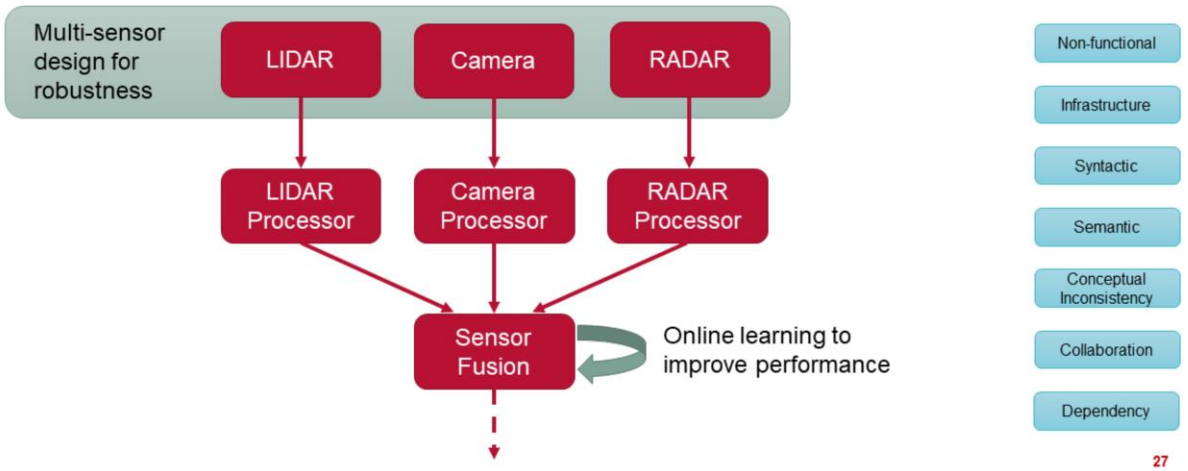
SYSTEMS AND ENGINEERING TECHNOLOGY

- In this version of the example the system designers were aware of concept drift and developed their perception component to be able to learn during operation and recognise new classes of object.
- Unfortunately the same thought process was not applied to the prediction component so we encounter a conceptual inconsistency fault that wasn't present at design time as the components' understanding of the environment has diverged.



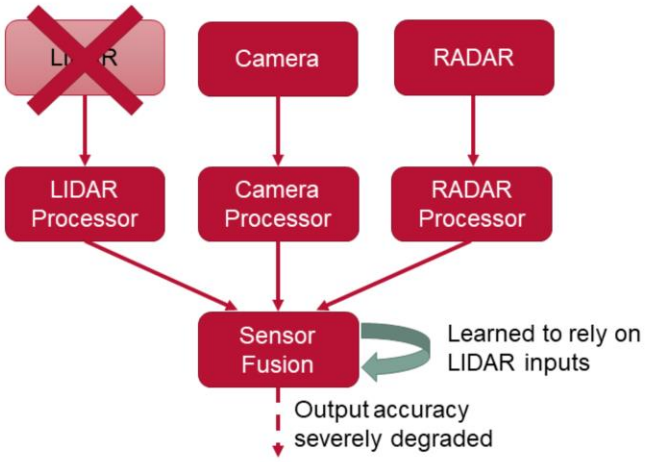
- Perhaps the system designers thought of this problem and included a means of new concepts being shared between the perception and prediction components.
- The means of defining a new concept could be quite complex and that risks introducing collaboration faults during operation. This is due to the environmental variation being unknown at design time (otherwise online learning would not be necessary), therefore the nature of concepts being shared is hard to predict and the component interactions difficult to adequately assure.

Example 4 - Adaptation



- Here we have a different example system that fuses information from multiple sensors to increase robustness and capability in a range of environmental conditions.
- The sensor fusion component is adaptive to improve performance throughout the operation of the system

Example 4 - Adaptation



Fault Type

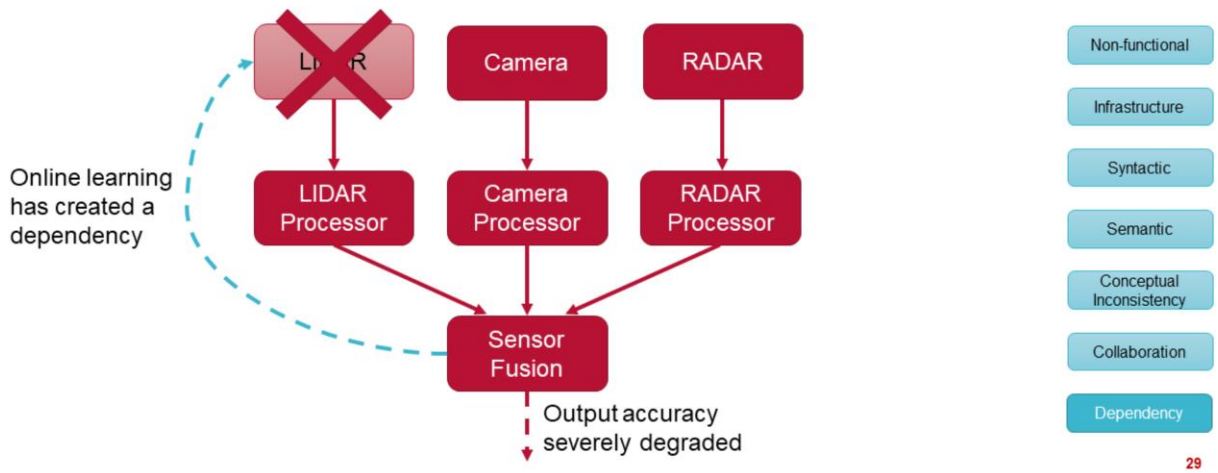
- Non-functional
- Infrastructure
- Syntactic
- Semantic
- Conceptual Inconsistency
- Collaboration
- Dependency

28

SYSTEMS AND ENGINEERING TECHNOLOGY

- Over time the sensor fusion component has learned to rely on LIDAR inputs as it has determined that they provide the highest accuracy.
- During operation the LIDAR sensor is unavailable for some reason, for example equipment failure or unusual environmental conditions.
- Consequently the output of the sensor fusion component is severely degraded impacting overall system behaviour.

Example 4 - Adaptation



SYSTEMS AND ENGINEERING TECHNOLOGY

- At run time, the online learning has created a dependency on LIDAR that was explicitly designed out during development. This poses a significant assurance challenge as the system designer's intent is eroded and then contradicted over time.

Next Steps

- Finalise fault taxonomy
  - More work to do on dynamic aspects
  - Feedback welcome and encouraged! Email c.allsopp@fnc.co.uk
- Understand characteristics and identify root causes of faults
  - Spoiler alert: specification issues will be prevalent
- Identify techniques to prevent and check for faults
  - Likely to be gaps - help define future research
- Try the approaches on example systems
- Publish our findings

30

SYSTEMS AND ENGINEERING TECHNOLOGY

- To conclude, there's more work to do on refining the taxonomy, particularly for the dynamic cases that introduce additional complexity into autonomous systems.
- As mentioned we'd be delighted to receive feedback or discuss our approaches. Please email me in the first instance and I can connect the team together.
- We'll then move on to understanding the characteristics of faults and identify root causes. We suspect specification weaknesses and implicit assumptions will feature heavily, which aligns with the known challenges around specifying autonomy and artificial intelligence.
- The focus will then be on identifying the assurance techniques that are available at varying levels of maturity to prevent or detect the issues we've identified. We'll also investigate how combinations of complementary techniques can build increasing confidence. There will almost certainly be gaps and suggestions for future research themes.
- We'll also attempt to validate some of the promising approaches on example systems.
- Finally, we'll hopefully have some findings that are beneficial to the community. If so, we'll look to publish through a range of channels.


FRAZER-NASH CONSULTANCY
University of BRISTOL
dstl
UWE Bristol
University of the West of England
Thank you. Any questions?
Feedback welcome!
Chris Allsopp – Group Leader, Autonomous Systems Engineering
c.allsopp@fnc.co.uk
www.fnc.co.uk