KUGLER MAAG CIE NA

MBD and ASPICE Compliance

Spring 2022

Peter Abowd / Steve Tengler Kugler Maag Cie NA Inc.

KUGLER MAAG CIE

Top Questions Encountered

Kugler Maag Cie performs dozens of assessments annually

- In 2021 alone > 250 assessments globally
- More than half of those included some form of model-based development
  - For design aspects alone
  - For design and code generation
- In this session, focused solely upon MBD for embedded software
  - Not addressing ASPICE SYS.2 or SYS.3

In this session we will discuss the following:

- Challenges with Model Based Software Development
  - Is it Architecture, Design or Implementation?
  - How do MIL, SIL/PIL assist compliance?
  - What is Detailed Design and what can be a Unit?
- Considerations for an Effective Strategy

- Cannot teach software design in 15 minutes, but we may be able to convey critical items that can help improve designs and compliance

Product Development Confusion

Problems With The Way We Talk

General confusion on what we are talking about

- Problem Vs. Solution
- What Vs. How
- Requirements (Features) Vs. Architecture/Design (Functionality)

Features = A Table of Contents

- Outline of detailed Requirements
- Defines what the product should provide to the end User
- May include a ***constraint*** on How to implement – should be separate

Functionality = What Product Engineers design/implement

- Features are realized though the combination and coordination of explicitly engineered functionality that is designed, implemented and packaged into a single **Product** or a **Subsystem** of **Products**.

Overview of Architecting/Designing

…Describing What We Will Build…

The Confusion between **Problem vs. Solution** clouds the definition and purpose of Architecture/Design

- What is the "System" and Who is Designing it??
- How many "Levels"?
- Which terms, abstractions and views should be constant across levels?

- How do I reason or express the company's product Assets containing IP, especially across products lines, within product lines or between these design "levels"?
- How do I easily identify critical pieces, safety pieces, reusable pieces or redundant pieces?
- Instead of the architecture giving the whole team a common purpose and roadmap, it often is a source of confusion

- ASPICE can add to this confusion as it uses "System" with its own purpose and meaning
  - It refers to the resulting work product of the project team that is the object of the assessment

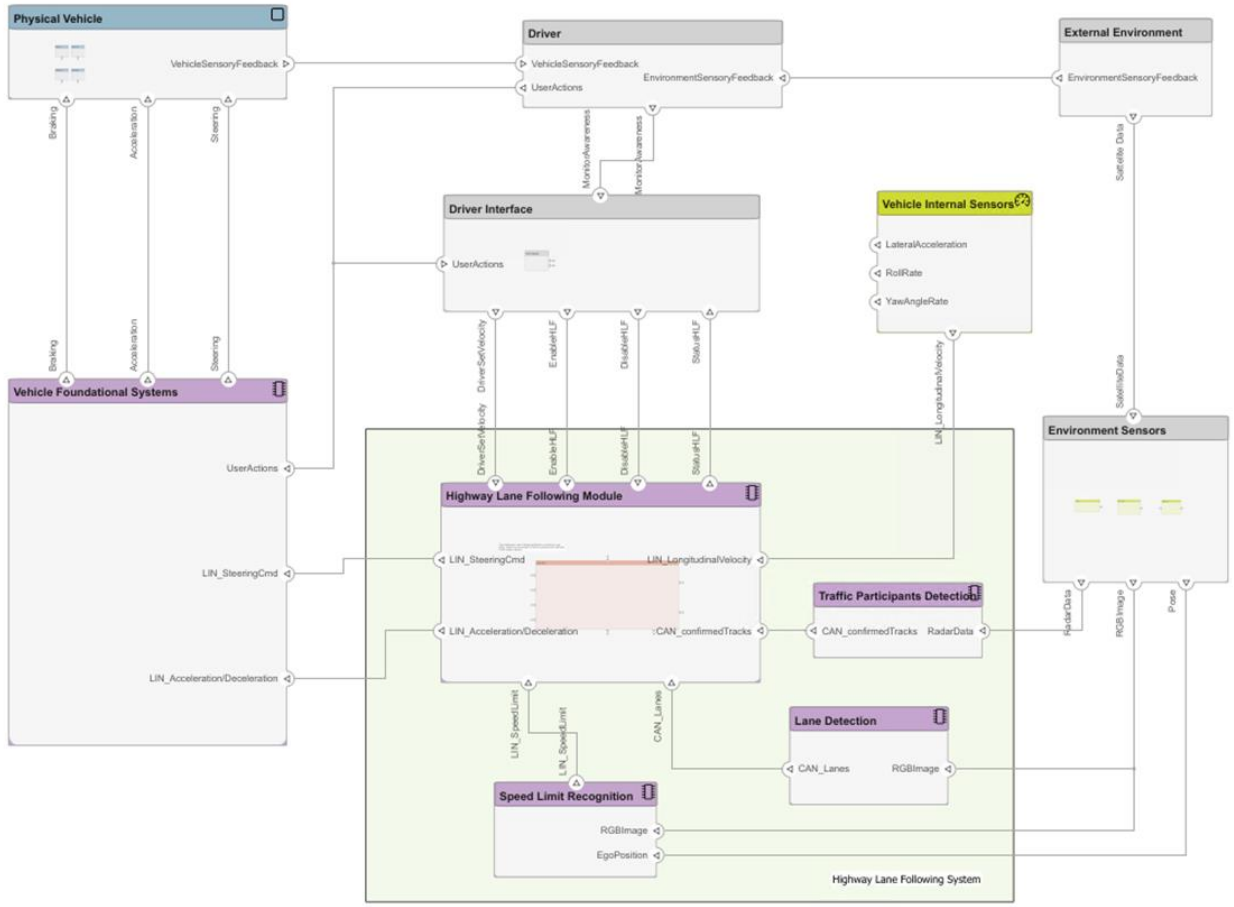
The Results

- Lost Business Opportunity to construct new products from Key Assets
- Lost Business Opportunity to manage profit through quality of Key Assets and Cost of (Re)Development
- Wasted Energy ($$) on managing all assets as equal or unknown criticality, priority…

Typical Compliance Challenges with MBD

Is the model and diagram sufficient for SWE.2?

- A single architecture diagram is not sufficient
  - Need rationalization / justifications of design choices
    - Why THESE pieces, and interfaces are more important (or not)
  - Different views with different semantic purposes
    - The model may not express the embedded runtime sufficiently
    - It also may not express ownership (make, buy, reuse etc.)
    - It doesn't convey performance, program or data size budget assumptions
- Must clarify where the Architecture ends, and Detailed Design begins
  - It is unlikely architecture is satisfactorily defined in one diagram
  - The Elements can be composed of Elements
- Semantic definitions are needed…for Example
  - What IS an "Element" of the architecture
  - Define motivations/rules for why creating Elements



Challenges

What Creates Bad Design

Bad Design is Bad Design, tool or no tool.

When to stop “Architecting”
and start “Detailing”?

What is a Unit, and why?

Bad Design

How to Identify It

- It doesn't make any more sense than the code does
- The semantics of the pictures are not defined or understood
- Design decision justifications are missing
- There are many pieces of design and no coordination
- There is a disconnect from the Architecture to the Detailed Design and then to the Unit(s)

comments

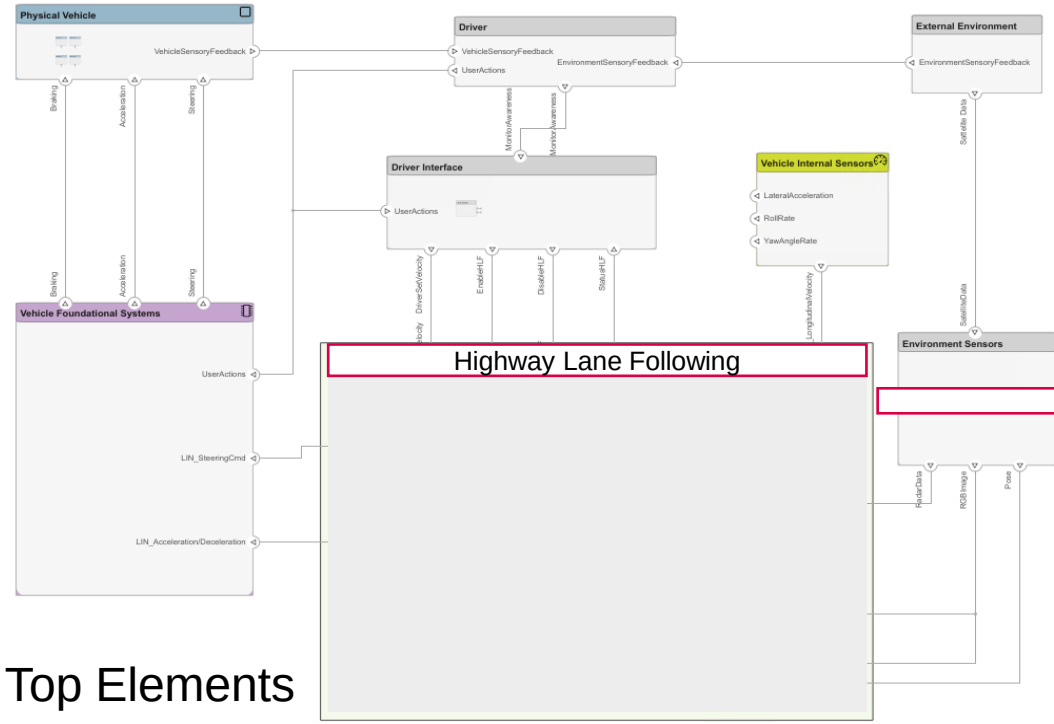
These can happen with or without the use of tools

These failures defeat the value of design, which is to provide a means of PREVENTING defects

This occurs by evaluating the design BEFORE implementation errors can be created

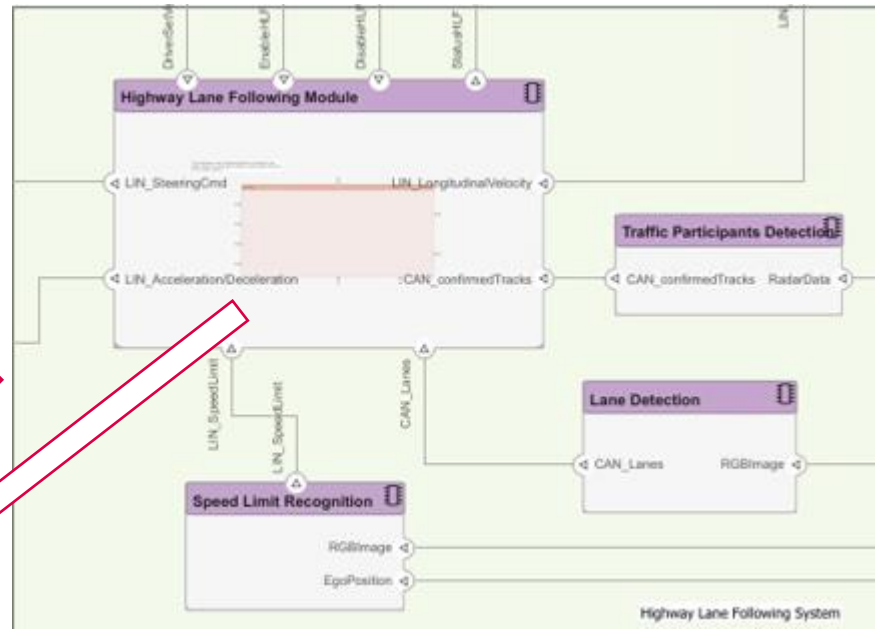
These common problems need to be addressed if your designs are to be effective

These items must be addressed by the project team; decided, documented and followed

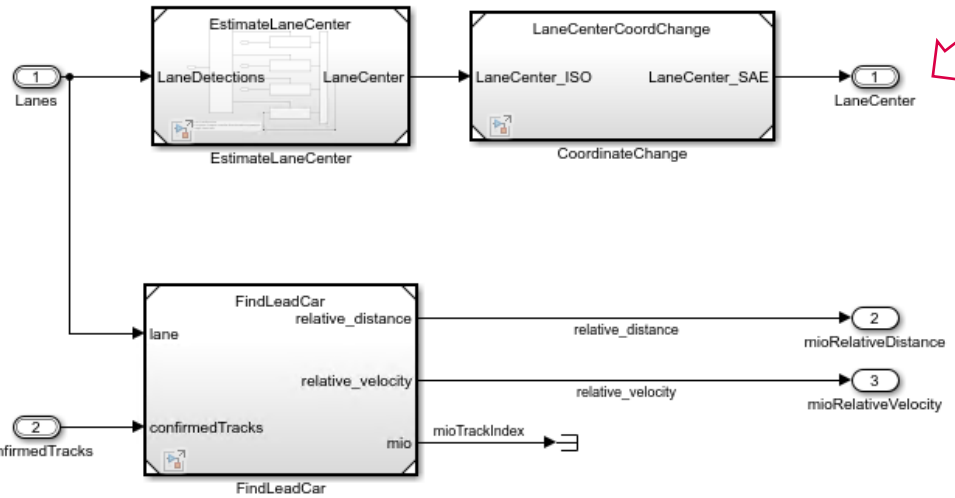


Top Elements

...

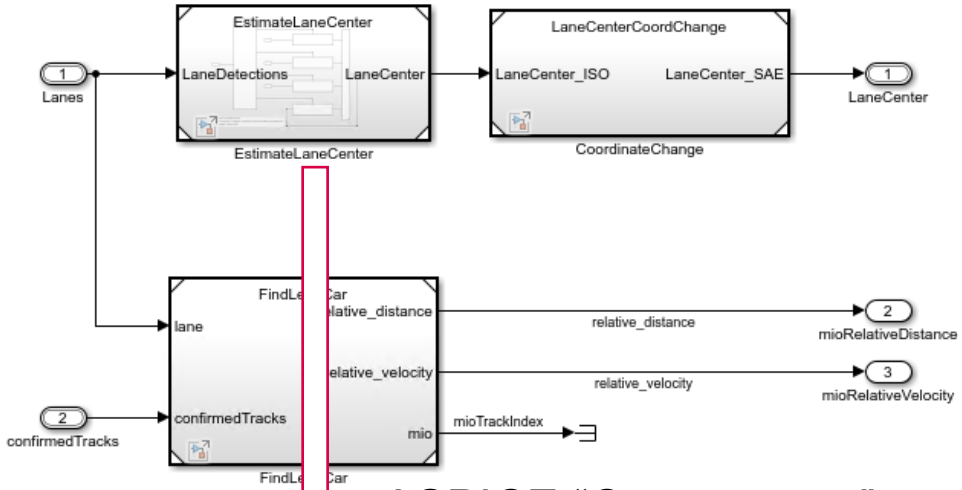


Decompose a Top Element into “sub” Elements

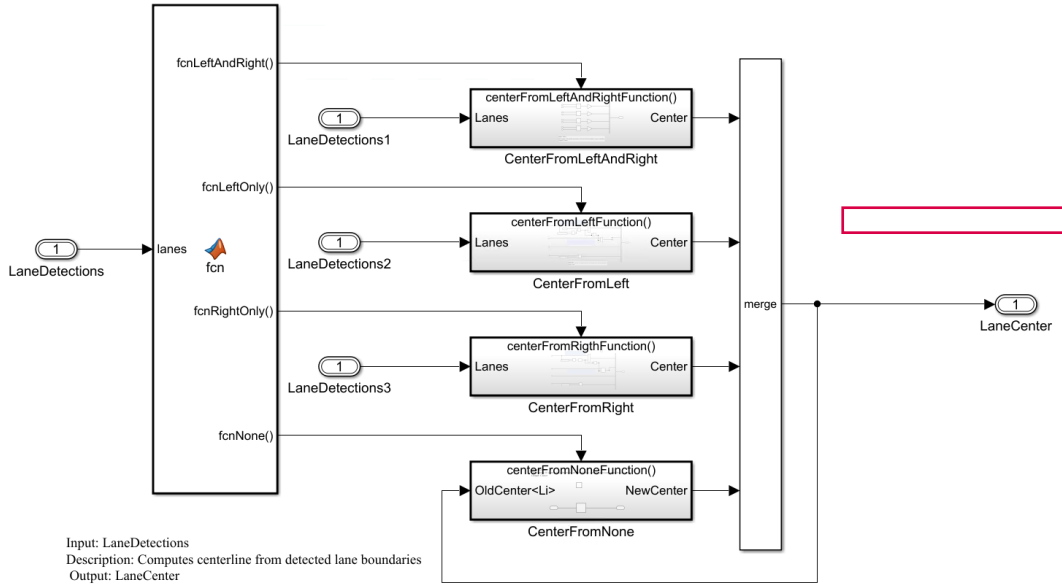


ASPICE “Components”

- The Details of a “sub” Element
  - Intended to be the end of architecture
  - These pieces are “atomic” end of architecture
    - In ASPICE they are “Components”
    - Will have a Detailed Design and implemented in Units
  - Making these pieces Model References is helpful



ASPICE “Components”

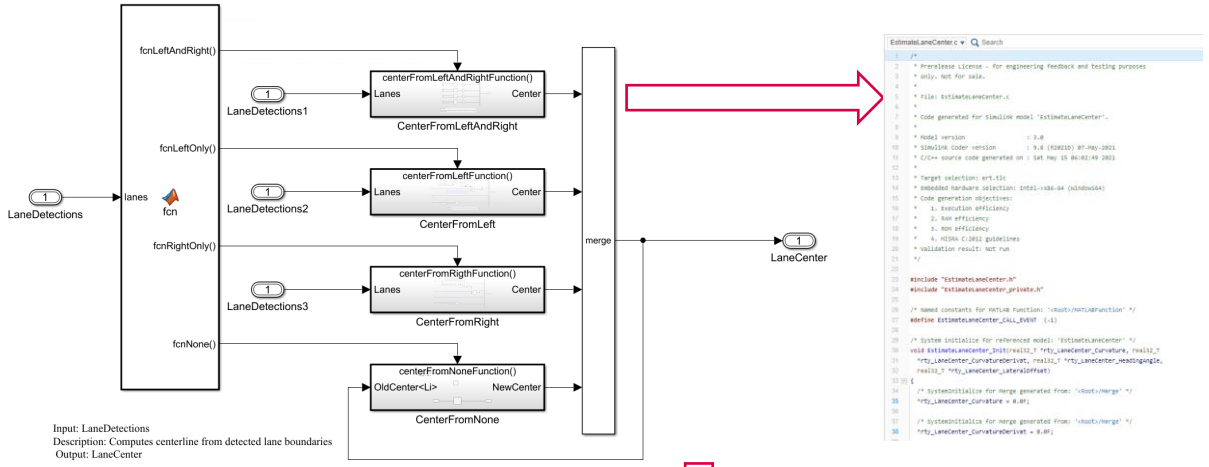


Detailed Design of “Component”

Model Reference Generated Code- A Unit

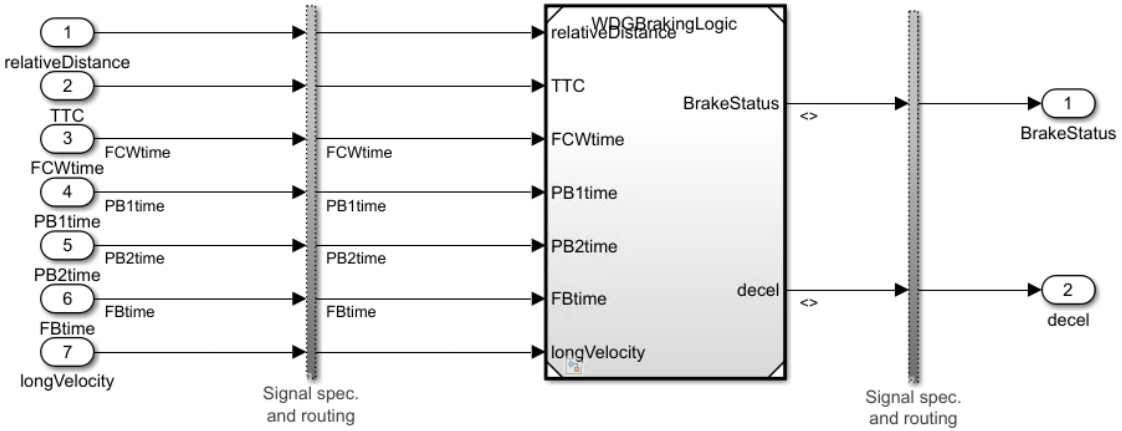
- The “Component” has its pieces too
  - In ASPICE a component requires a Detailed Design (DD)
    - In this case the model file for one of the “Components” (Shown at lower left)
- A Unit is source code implemented from a DD
  - In this case it is generated from the Model Reference (MR)
  - Using MR simplifies compliance
    - The DD is the MR file and the Unit the generated code

- This Unit requires Verification
  - Static & Testing



Detailed Design of “Component”

Generated Code- A Unit



Test Harness for a single “Unit”

The real power of MBD comes out now

- A test harness for the component / Model Reference
  - Used for MIL, SIL and PIL
  - Satisfies SWE.3 (MIL) for design evaluation
  - Satisfies part of SWE.4, Unit Test (SIL/PIL)
  - Need Back to Back results to fully satisfy unit testing
    - Success of SIL/PIL results compared to MIL
  - Perform Static analysis (machine based) for SWE.4
  - Human review of code can be argued away

The approach is very straight forward for compliance

- Also quite an effective engineering strategy
- Biggest risk is the introduction of MATLAB scripting
  - Often lacks design description
  - May provide significant functionality
  - If this is critical code, design information is needed

Summary

Summary Points

- ASPICE compliance does not mean significant change of behavior
  - Requires a strategy to meet Architecture, Detailed Design and Unit Verification compliance clearly
- A straight-forward approach has been outlined and proven to be compliant and helpful to teams
  - Defines some basic semantics and employs Model References.

Discussion

Peter Abowd
Peter.Abowd@kuglermaag.com
www.kuglermaag.us