System Architecture creation using System Composer

Sudeep Kulkarni, Engineer
MATLAB EXPO 2021

Mercedes-Benz
The best or nothing.

Agenda

1. Introduction
2. Problem statement
3. Solution
    a. System Composer – Introduction
    b. How can System Composer solve the problem?
4. Tools used
5. Demonstration
6. How did System Composer help us improve?
7. Conclusion

Introduction

Sudeep Kulkarni

- 5+ years of experience in Automotive Industry.
- Software Development for the following functions in Powertrain topics:
  - Front axle Manual transmission (FSG)
  - Speed control and coordination
- Currently a part of software development for functional safety of Transmission software

Problem Statement

- The process we follow in requirement based software development, lacks the standard process of creating requirement specifications

- Currently, the requirement specifications used by the developer for software development might be in the form of flow diagram, doors requirements, hand drawing etc.

- In order to achieve full V-Cycle with respect to software development, the process for analyzing the requirement and creating a system architecture that can be used for software development is necessary

Solution

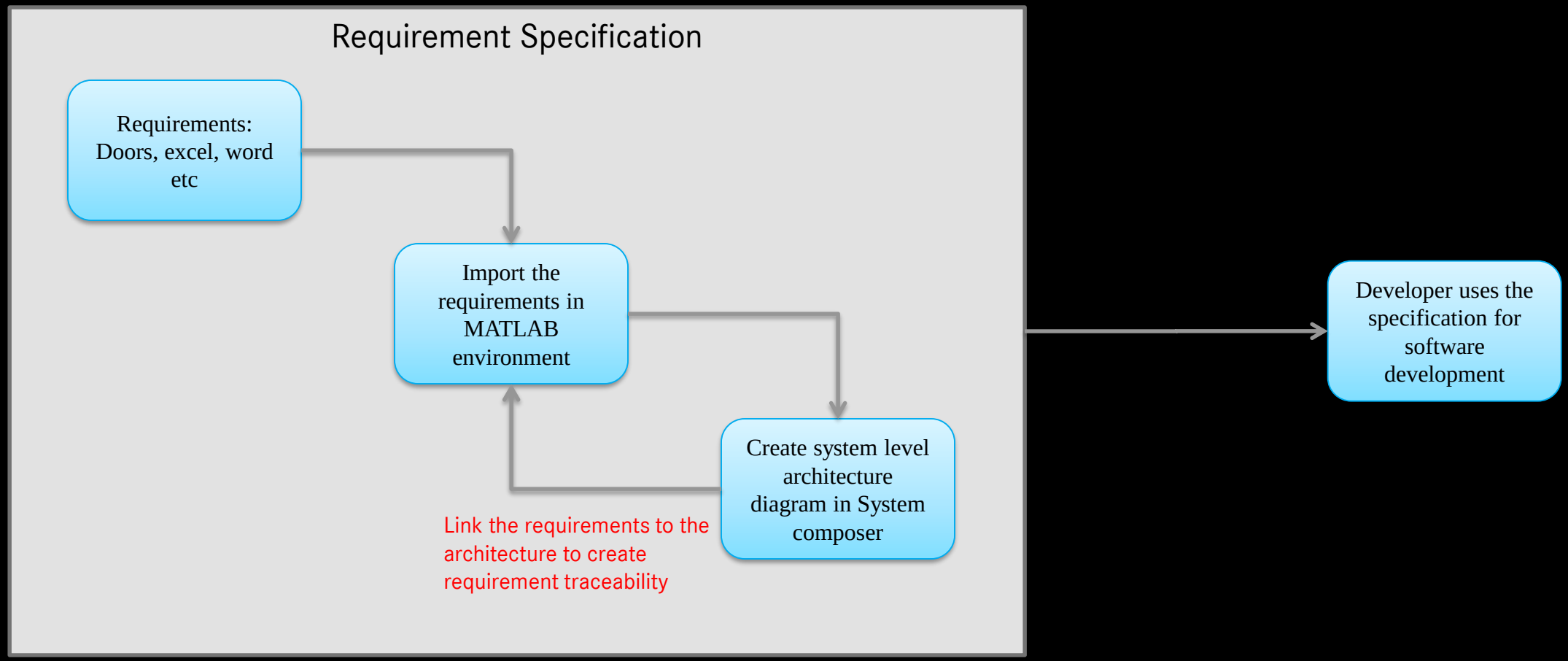
- To address this void, Architecture based software development shall be adopted with minimal changes in the conventional process

- With System Composer, this void could be addressed, as it is a toolbox in MATLAB family, the conventional process is not altered much

System Composer

- A MathWorks toolbox inside MATLAB family
- Import/create requirements using Simulink Requirements toolbox
- Enables user to create system level architecture
- Link requirements to the Architecture for better traceability
- Define stereotypes
- Create custom views
- Import Simulink model for system architecture creation
- And many more..

System Composer workflow - Proposed

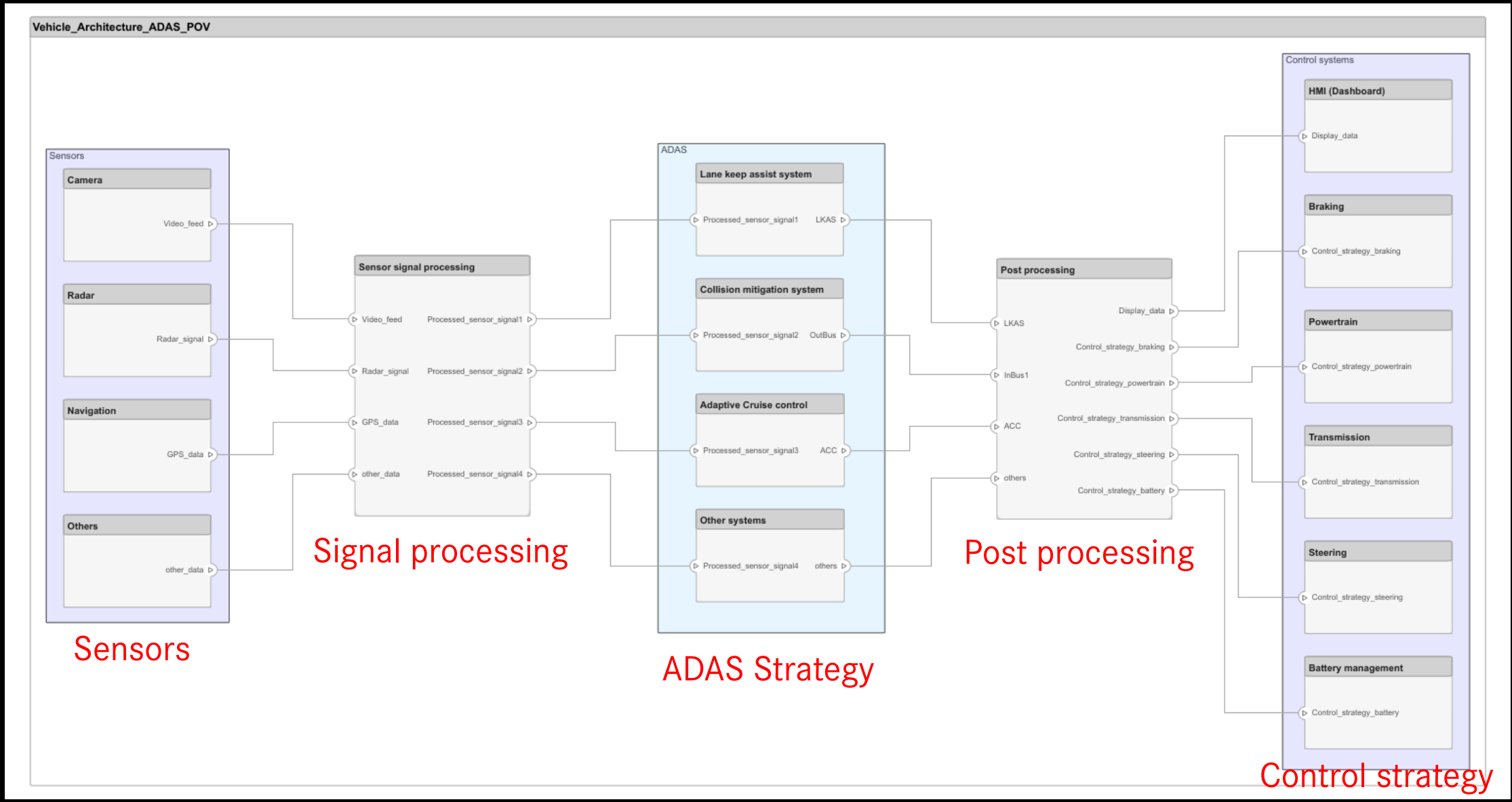


Tools used

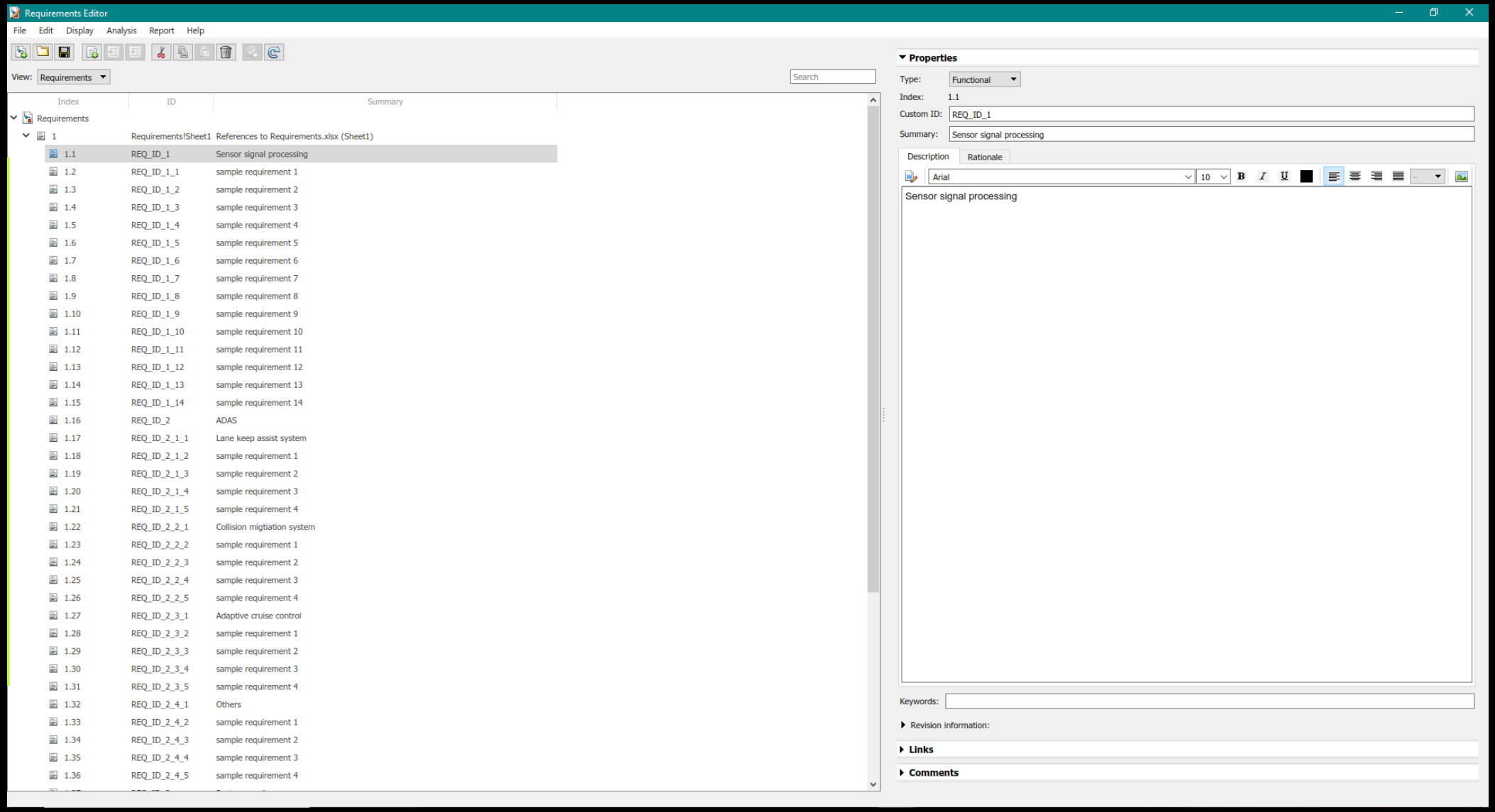
- Doors, excel, word etc – Requirements analysis
- Requirements import/export – Simulink Requirements Toolbox
- MATLAB 2020a System Composer Toolbox – System Architecture creation
- Other tools for software development

Demonstration of the tool

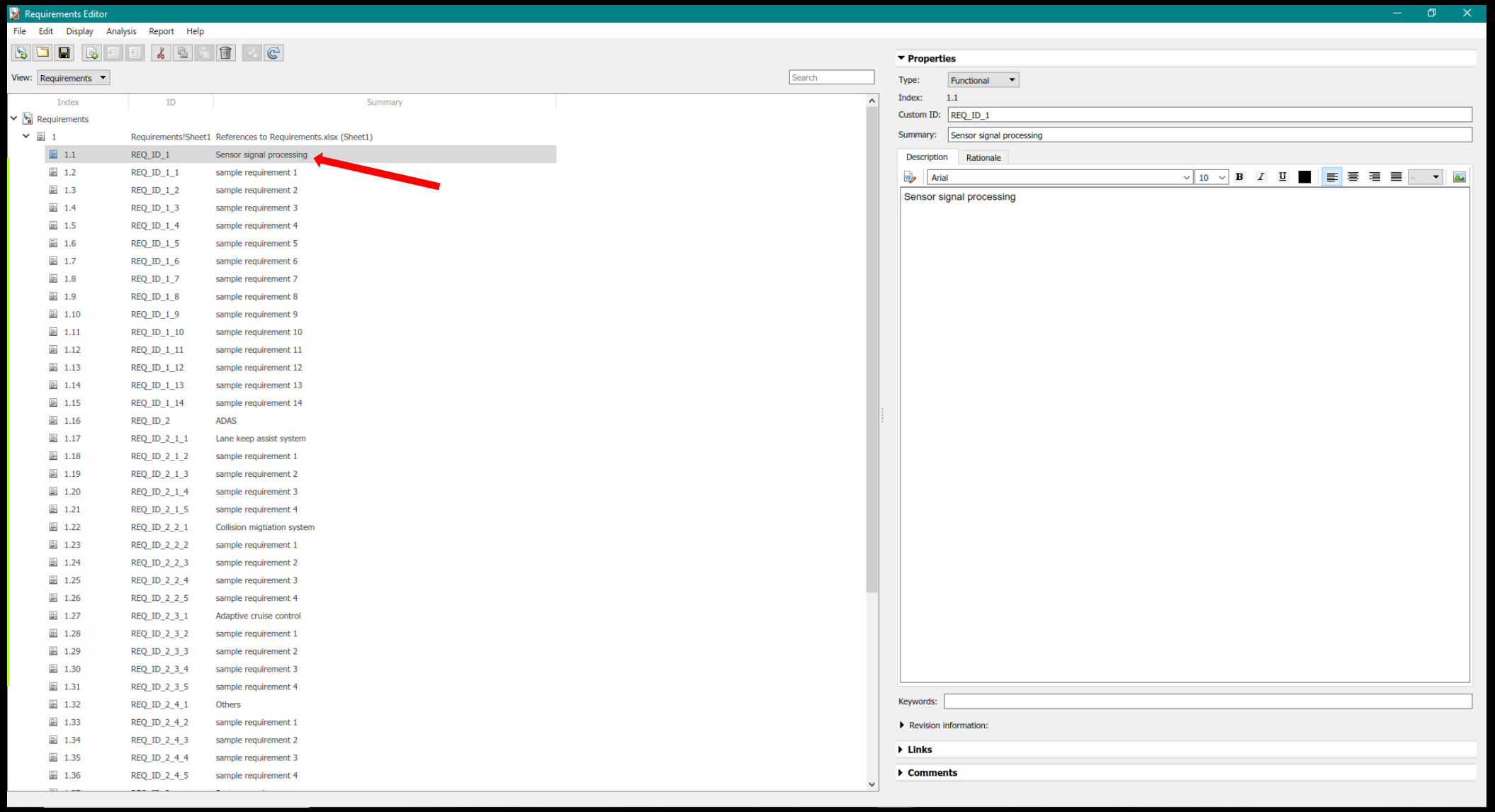
Vehicle Architecture from ADAS POV



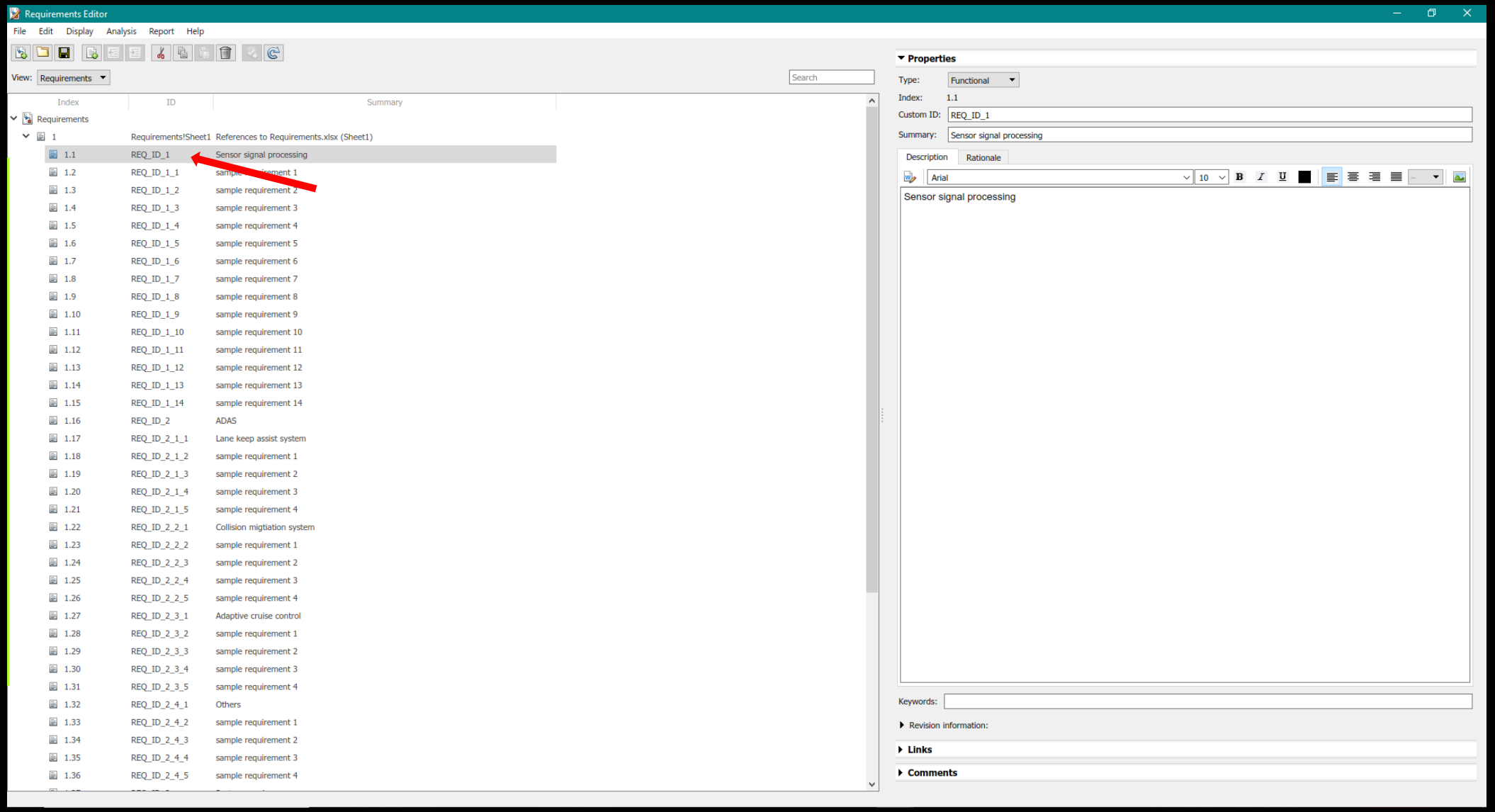
Requirements Editor



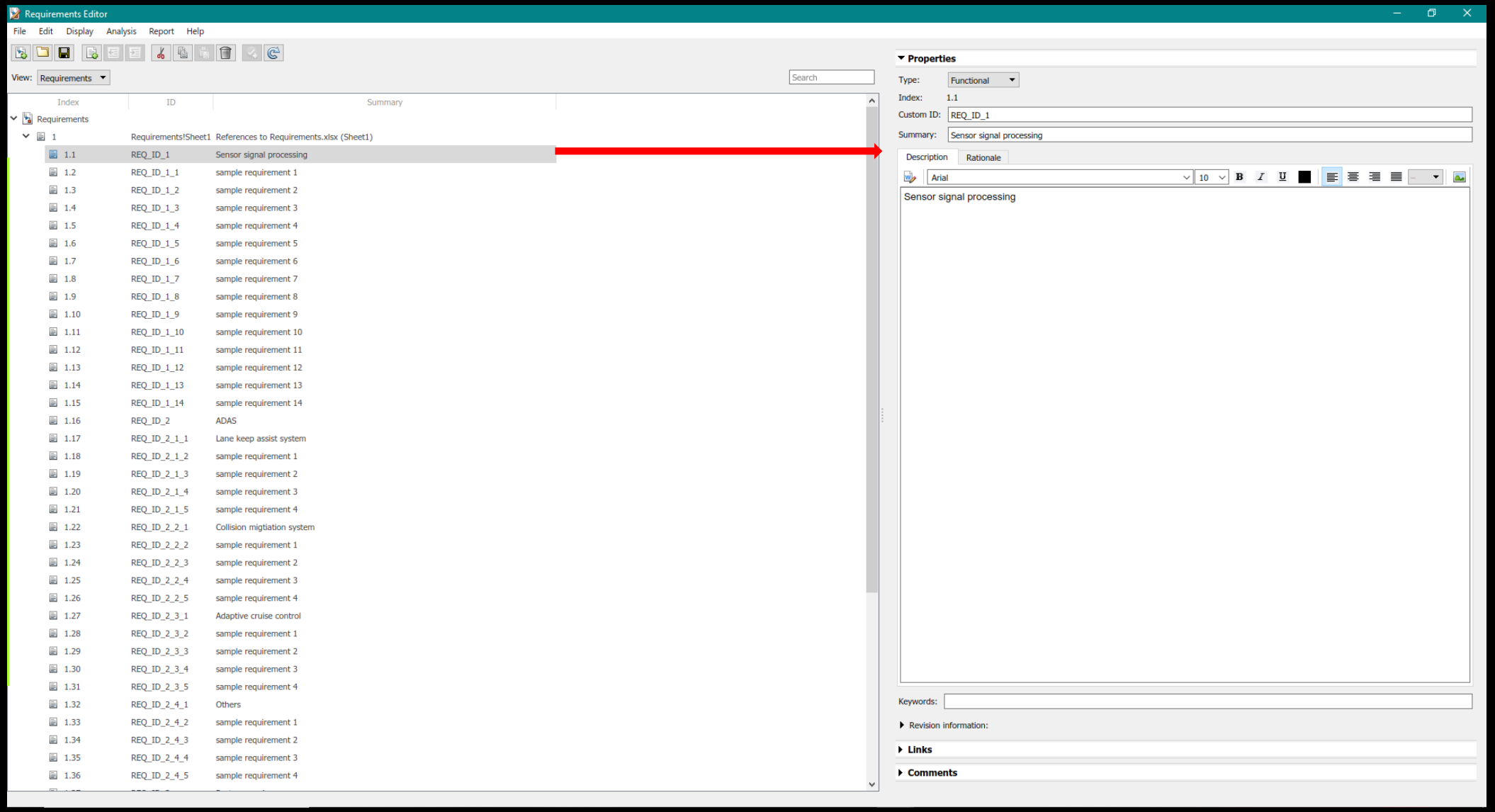
Requirements



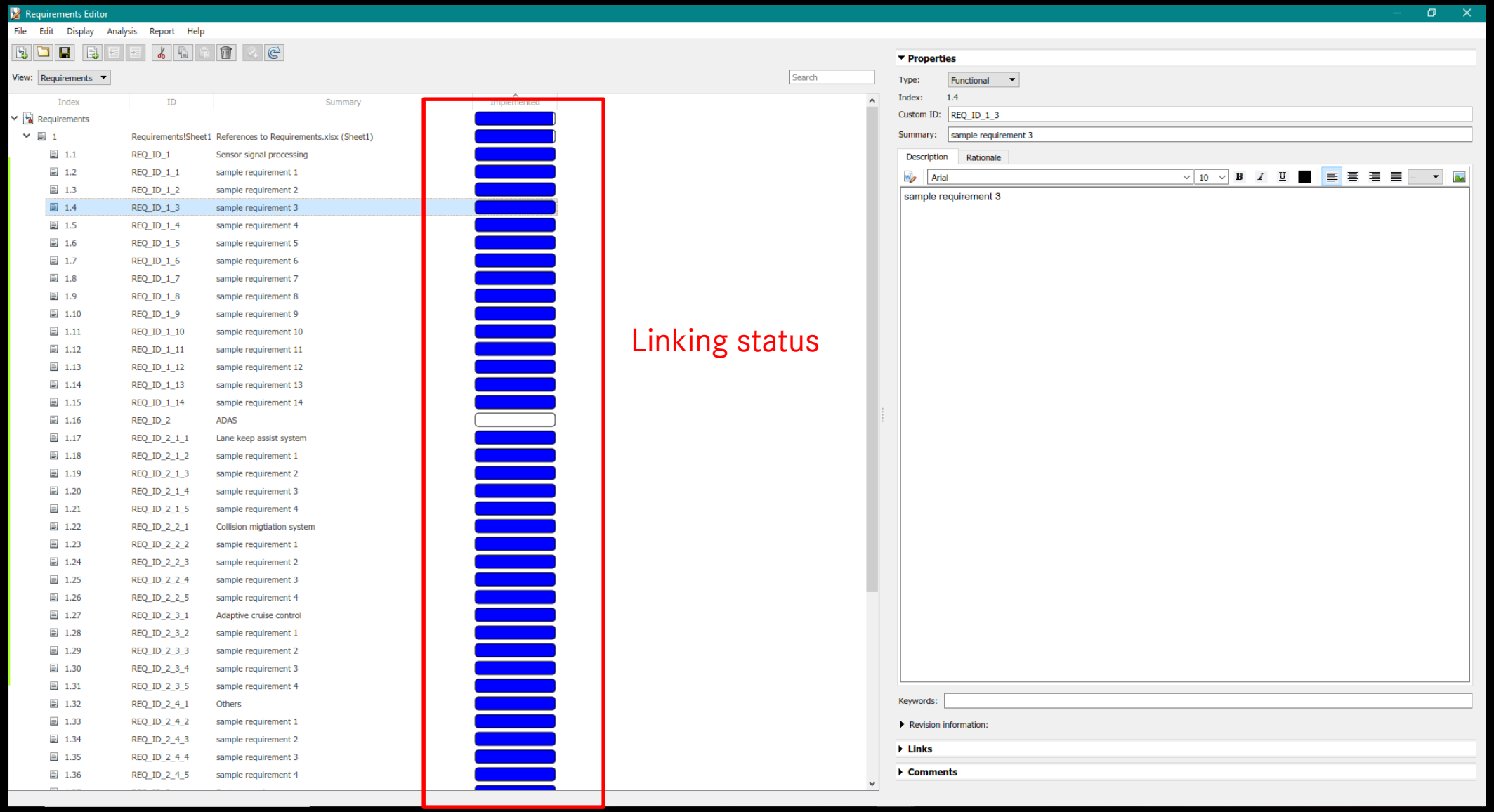
Requirements ID



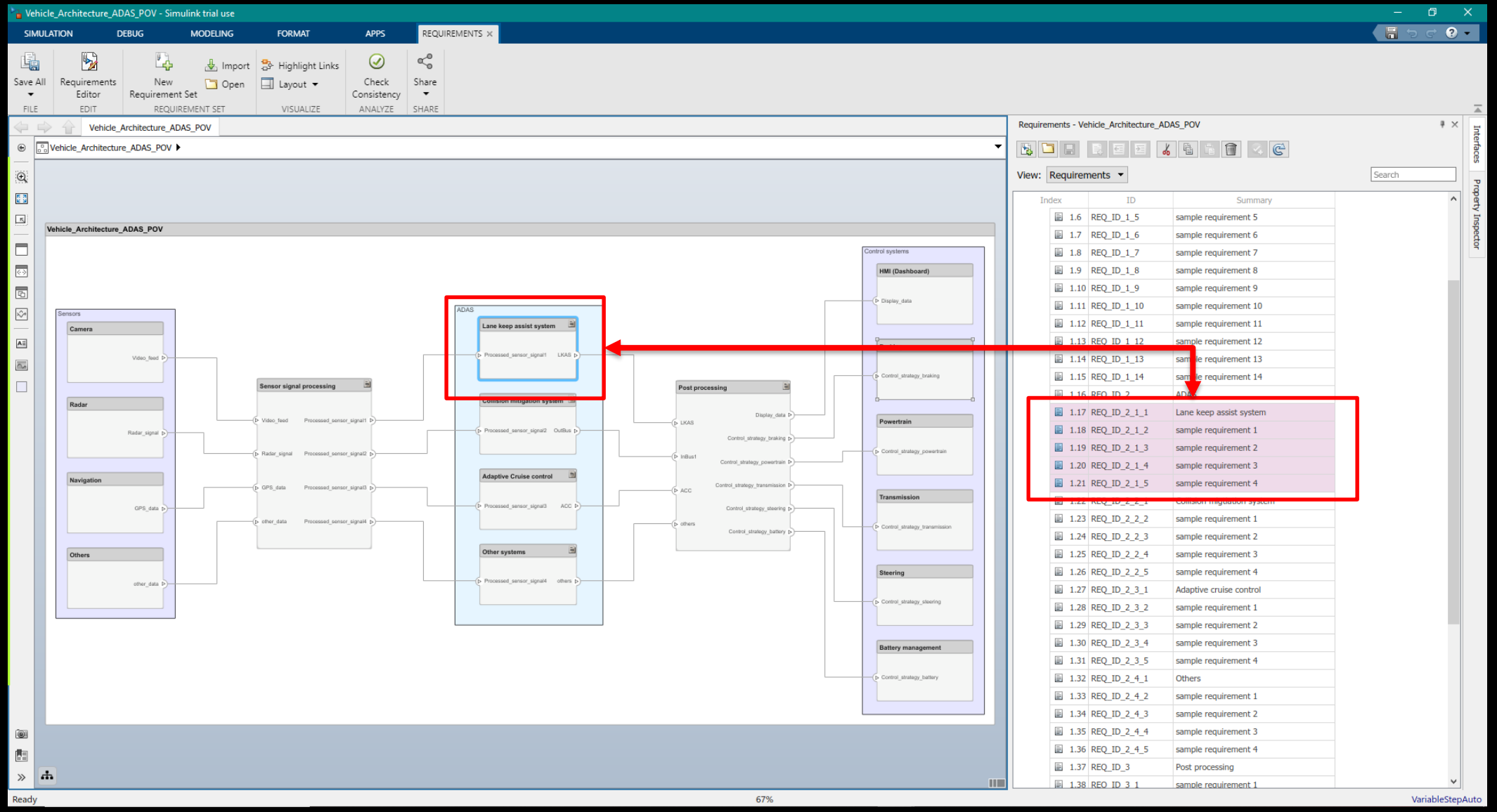
Requirements Properties



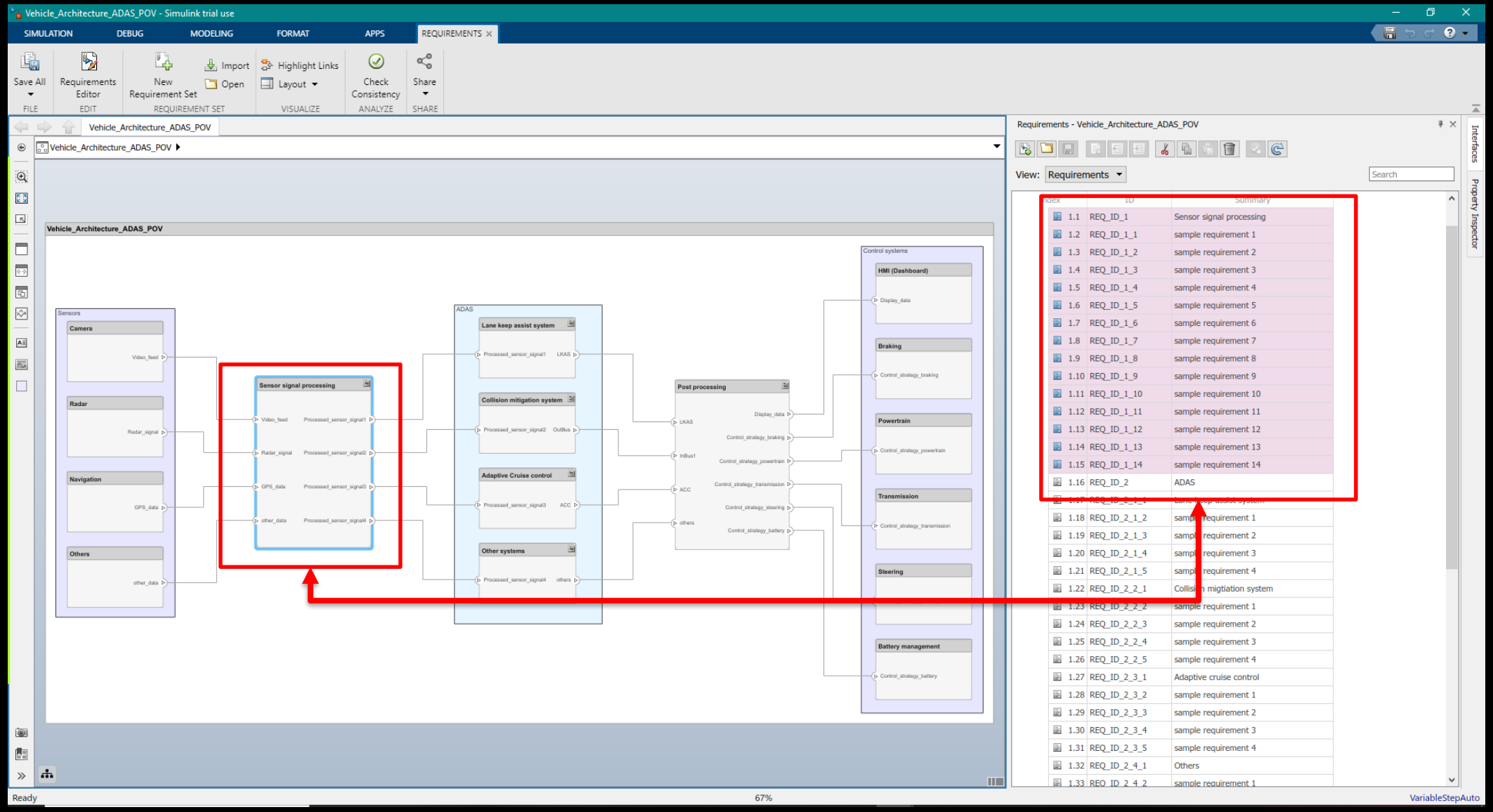
Requirements after linking to Architecture



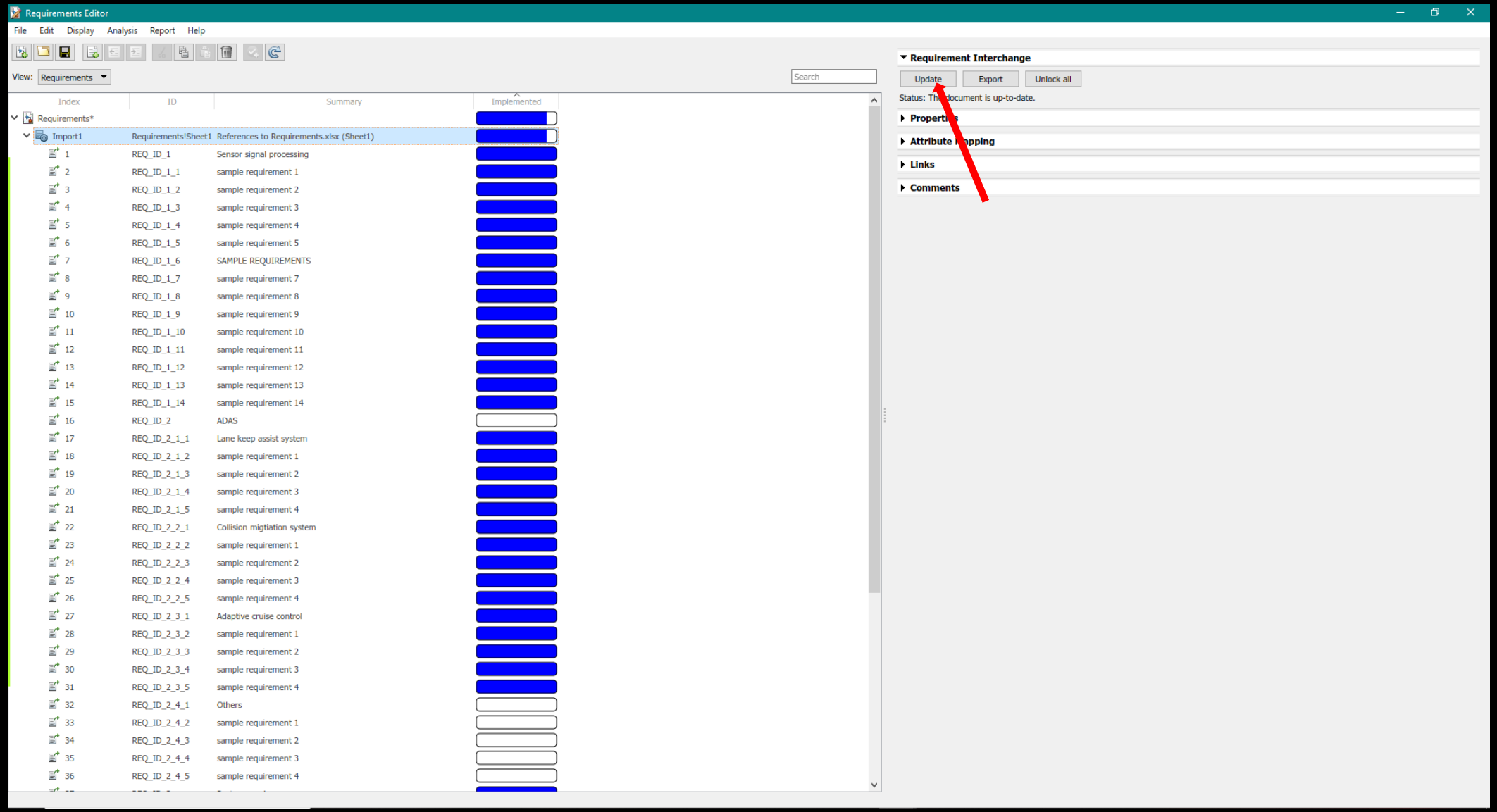
Requirements after linking to architecture



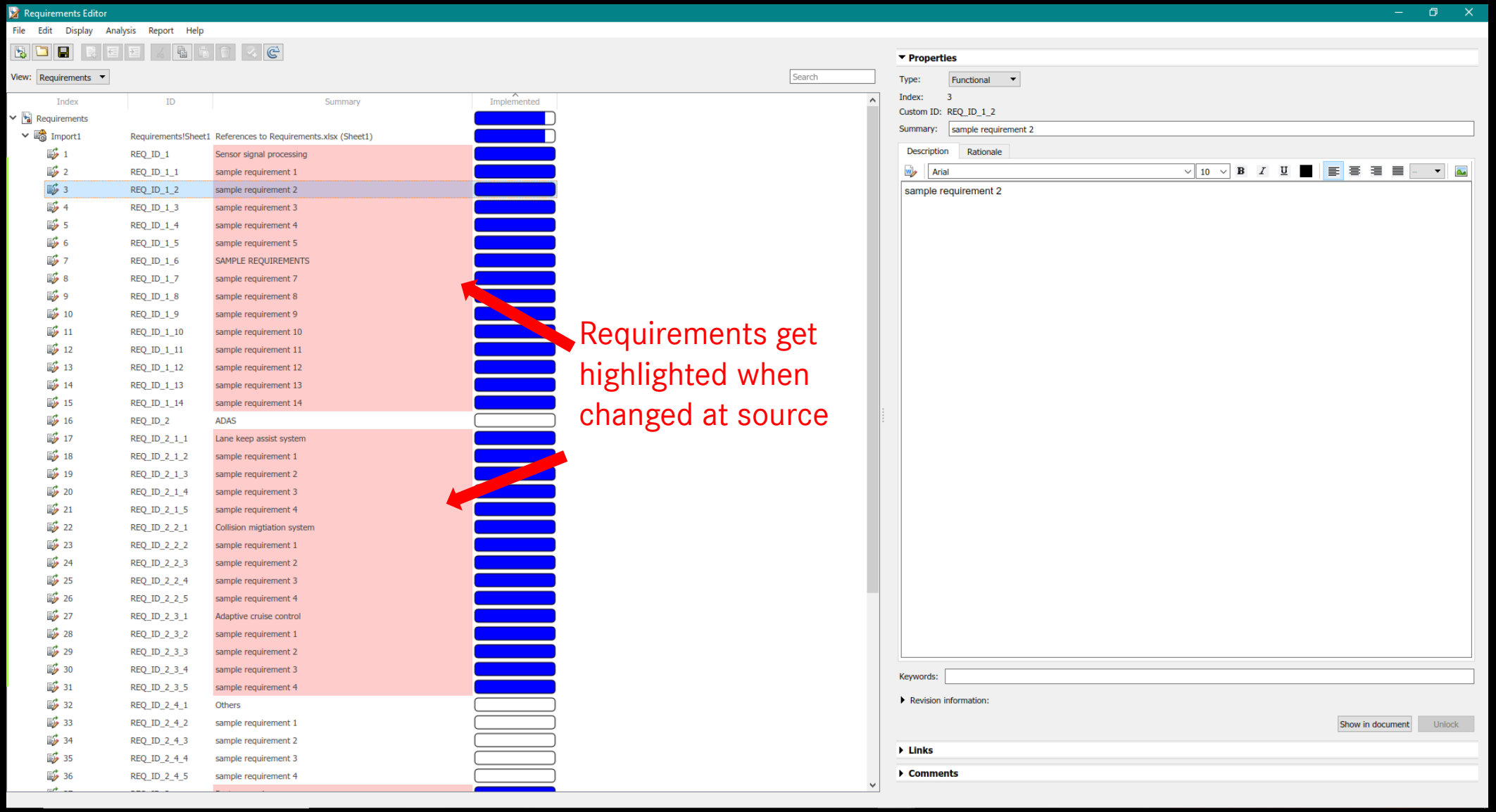
Requirements after linking to architecture



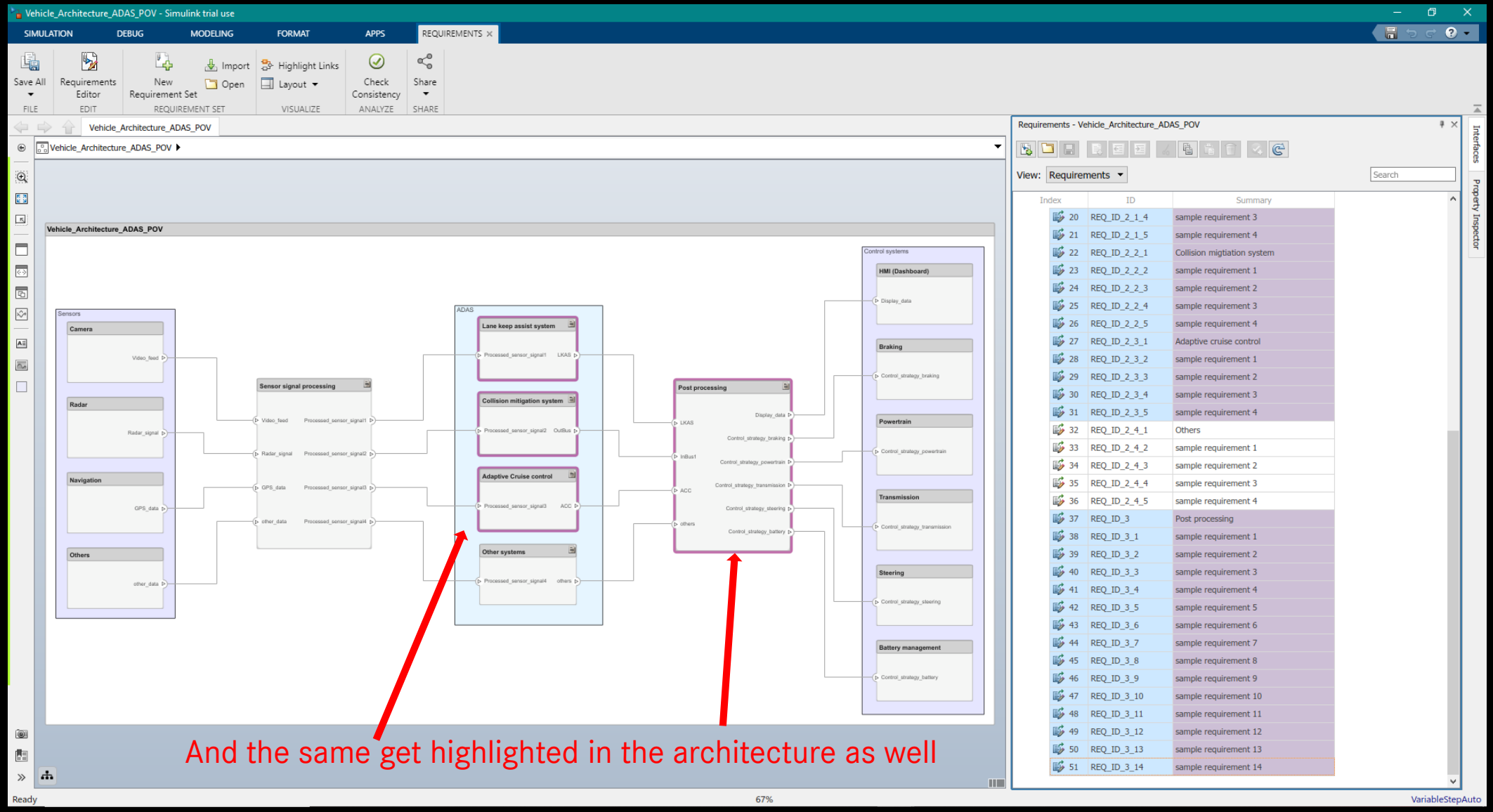
Requirements when updated at source



Requirements when updated at source

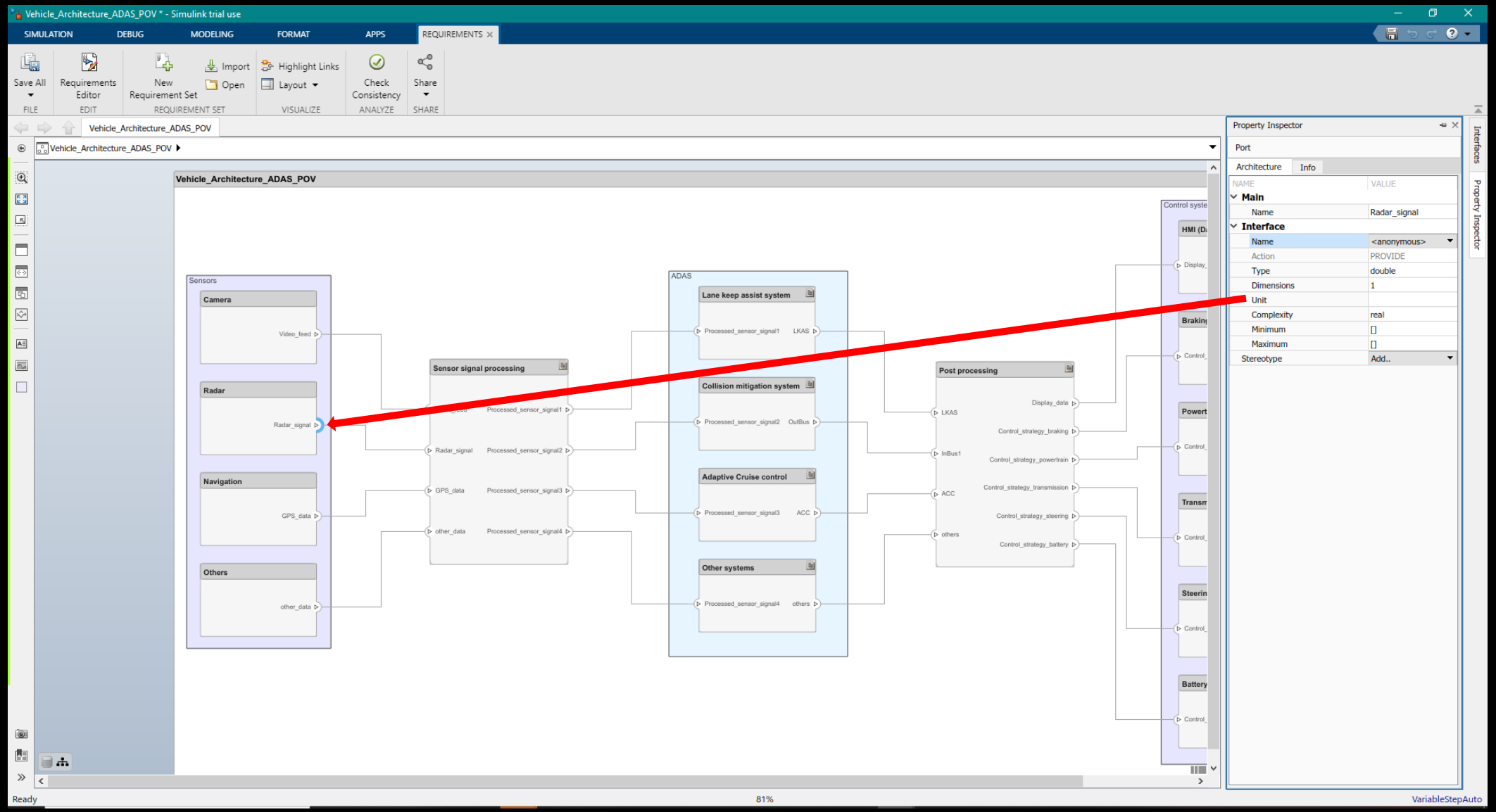


Requirements when updated at source

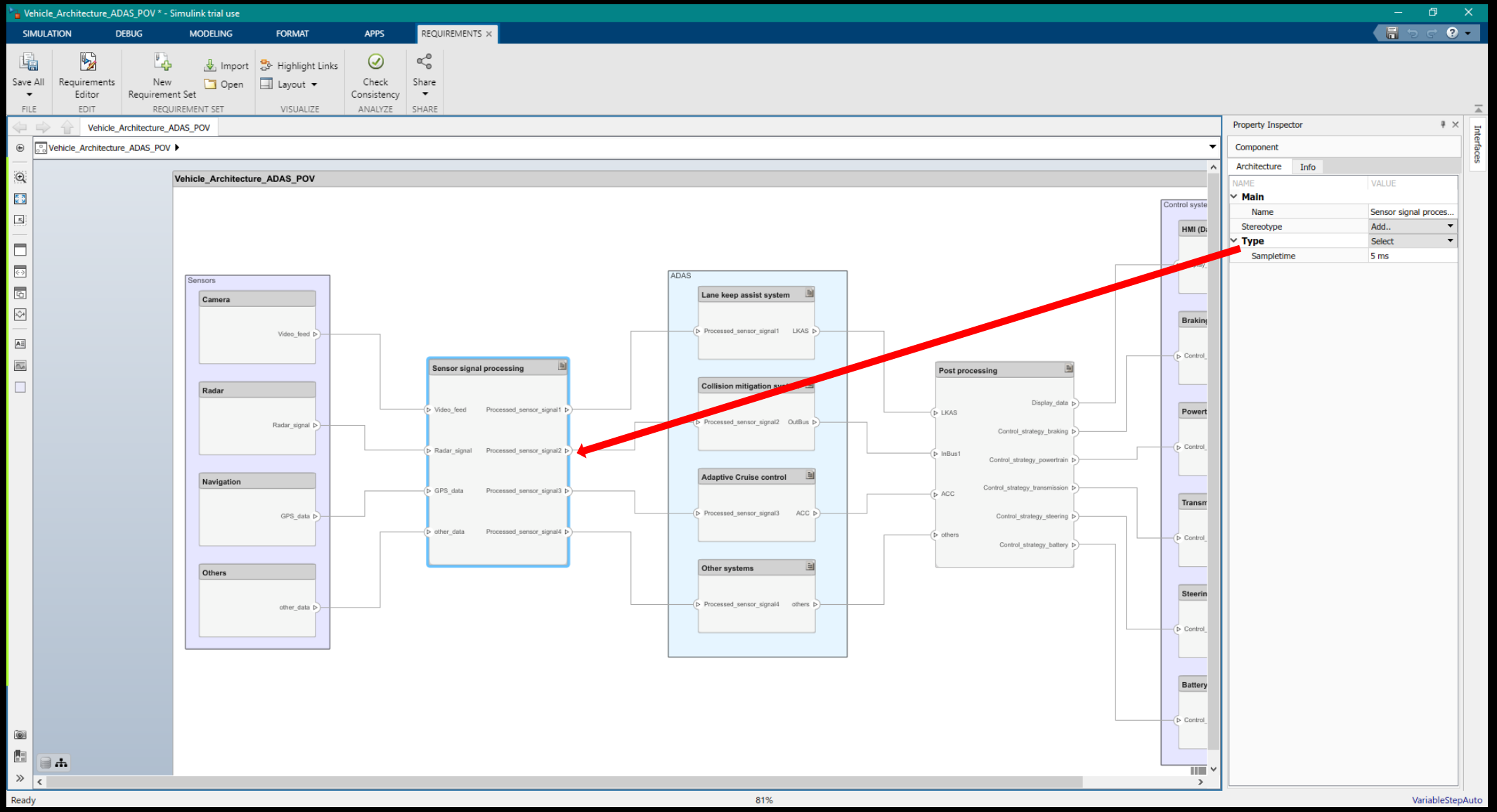


Some other features of System Composer

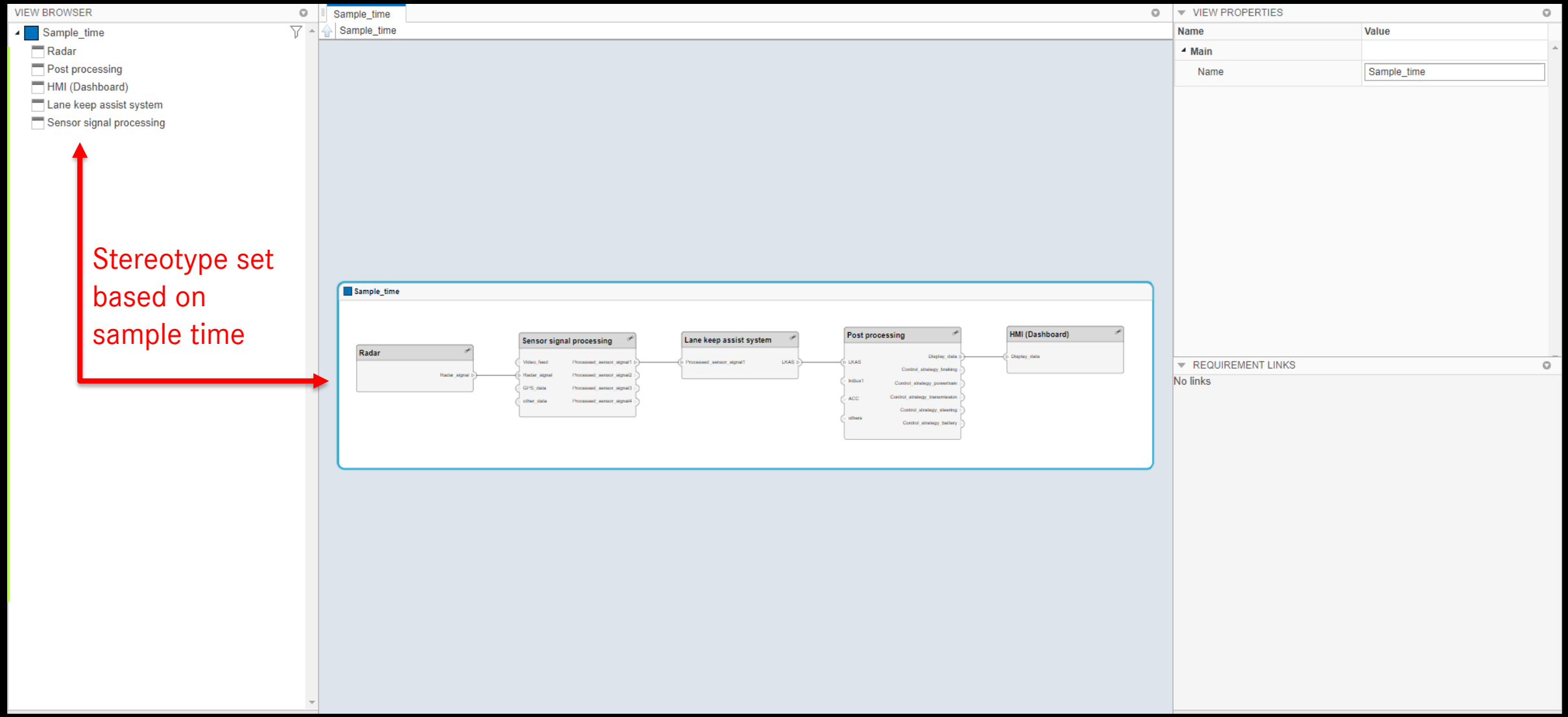
Interface data



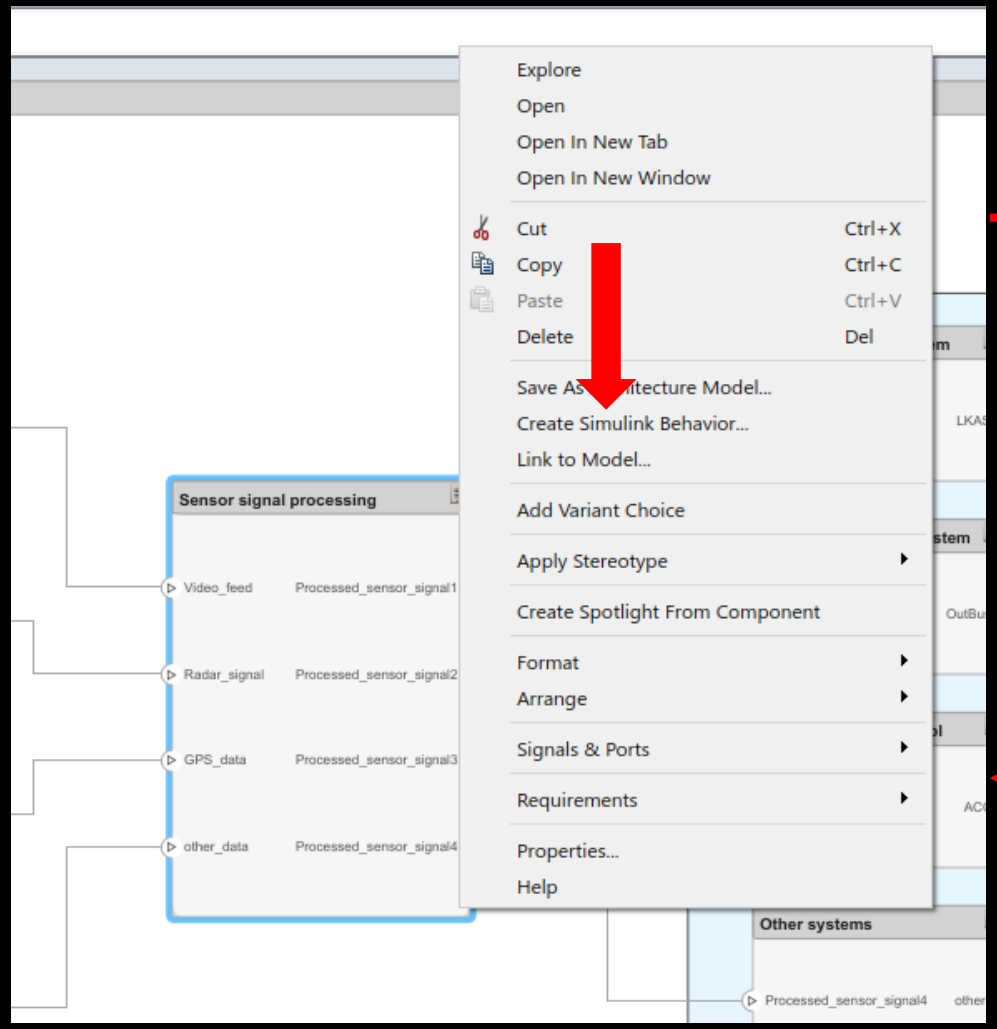
Stereotypes



Custom views

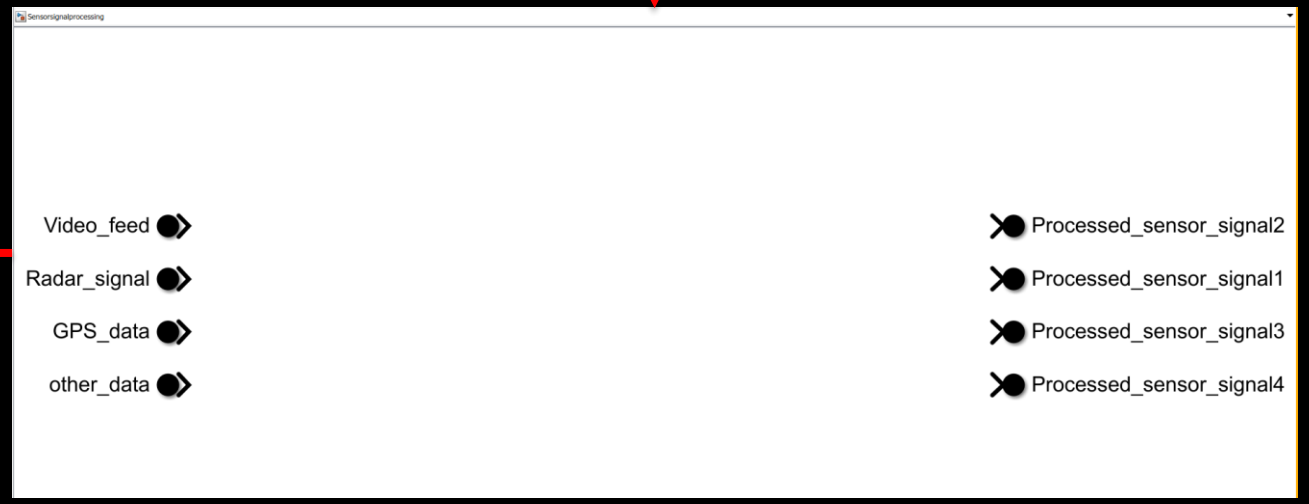


Create a Simulink behavior



System Architecture component

Any component with interfaces can be used to create a Simulink model with the predefined ports in the architecture.

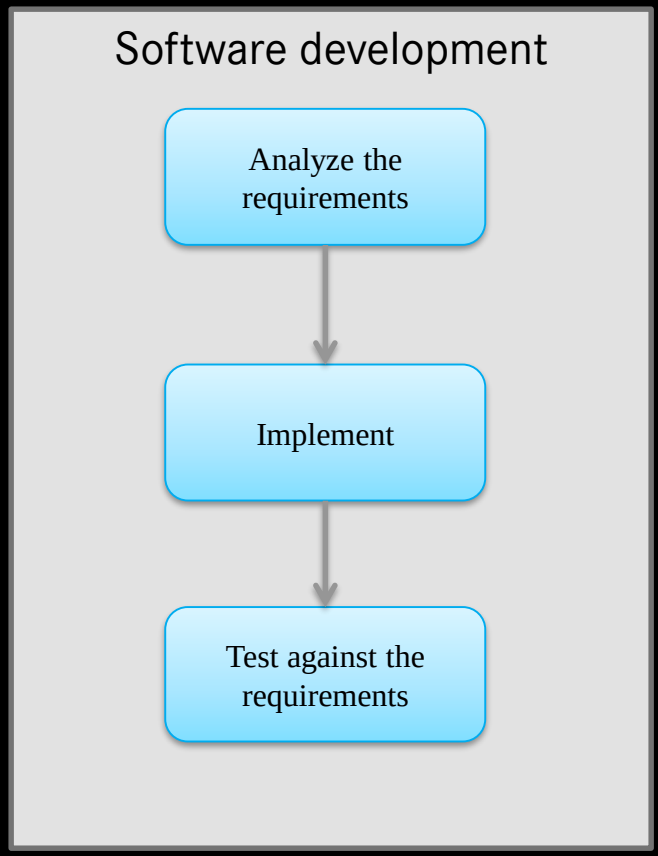


Simulink model

Other way around is also possible

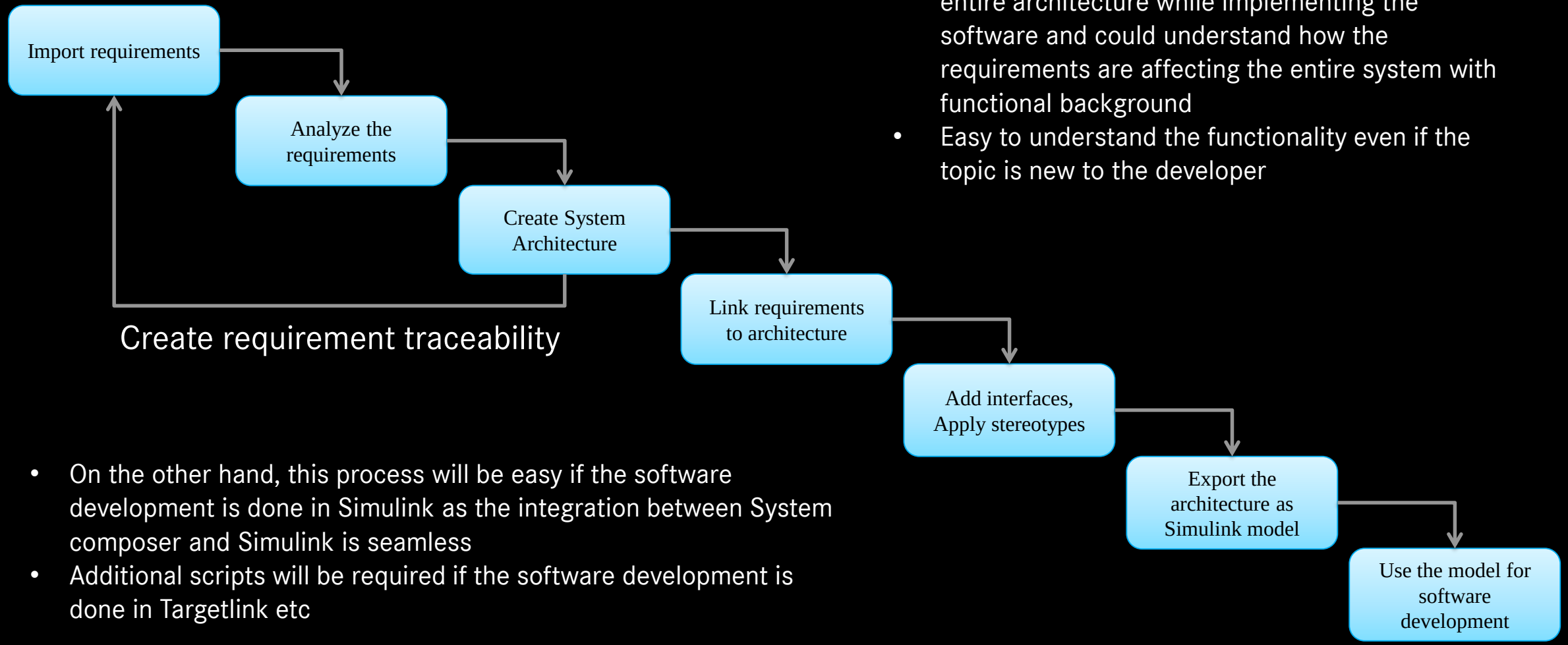
Requirements based Software Development

- The role of a software developer in the V-cycle of a Requirements based software development is limited.
- Mostly, analyzing the requirements, implementing it and performing unit test.
- What's missing here is that the developer won't understand the whole picture of the requirement being implemented on how it is affecting the entire system
- Might miss implementing some of the requirements due to human error
- Difficult to understand the functionality if the topic is new to the developer



Our findings after using System Composer

Process 1a: System to Software Architecture

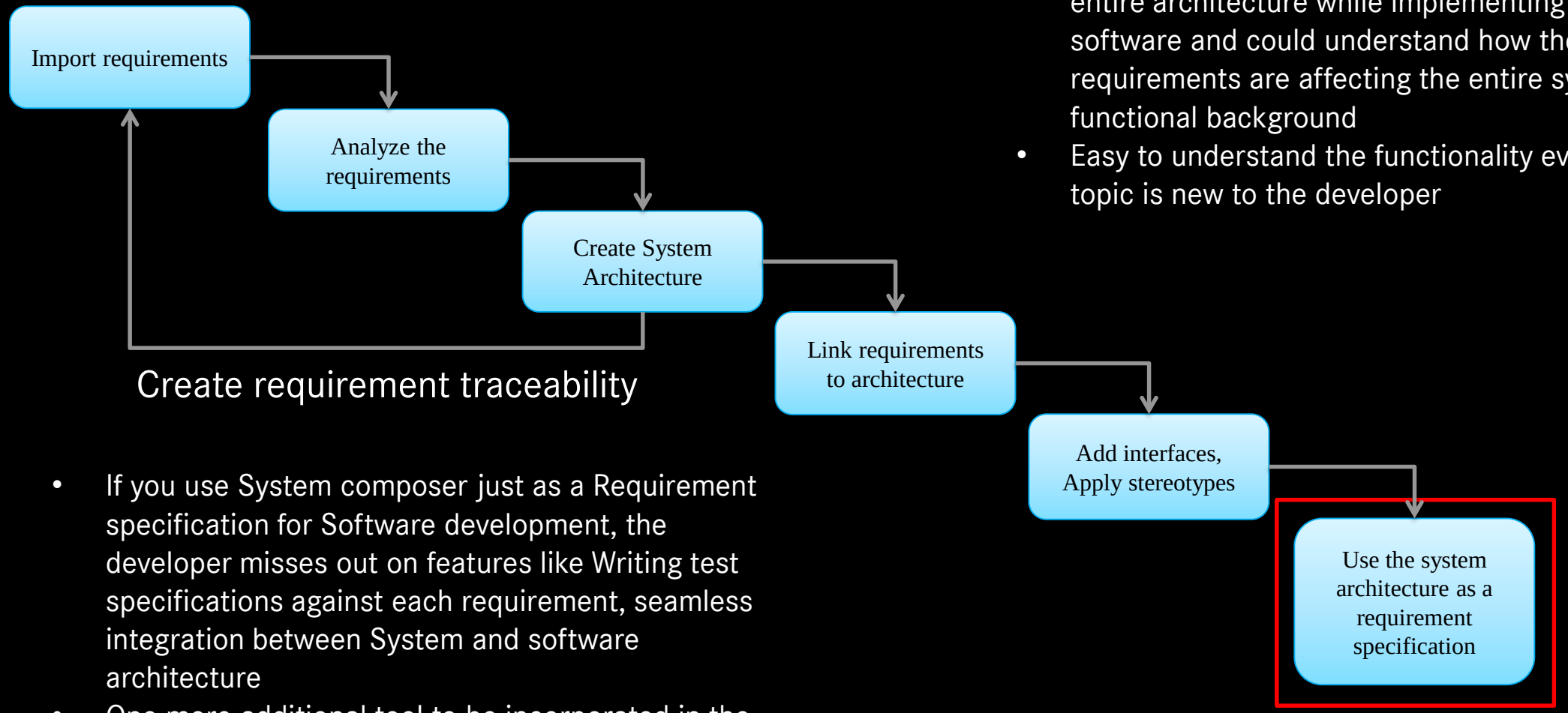


- In this process, the developer has access to the entire architecture while implementing the software and could understand how the requirements are affecting the entire system with functional background
- Easy to understand the functionality even if the topic is new to the developer

- On the other hand, this process will be easy if the software development is done in Simulink as the integration between System composer and Simulink is seamless
- Additional scripts will be required if the software development is done in Targetlink etc

Our findings after using System Composer

Process 1b: System to Software Architecture

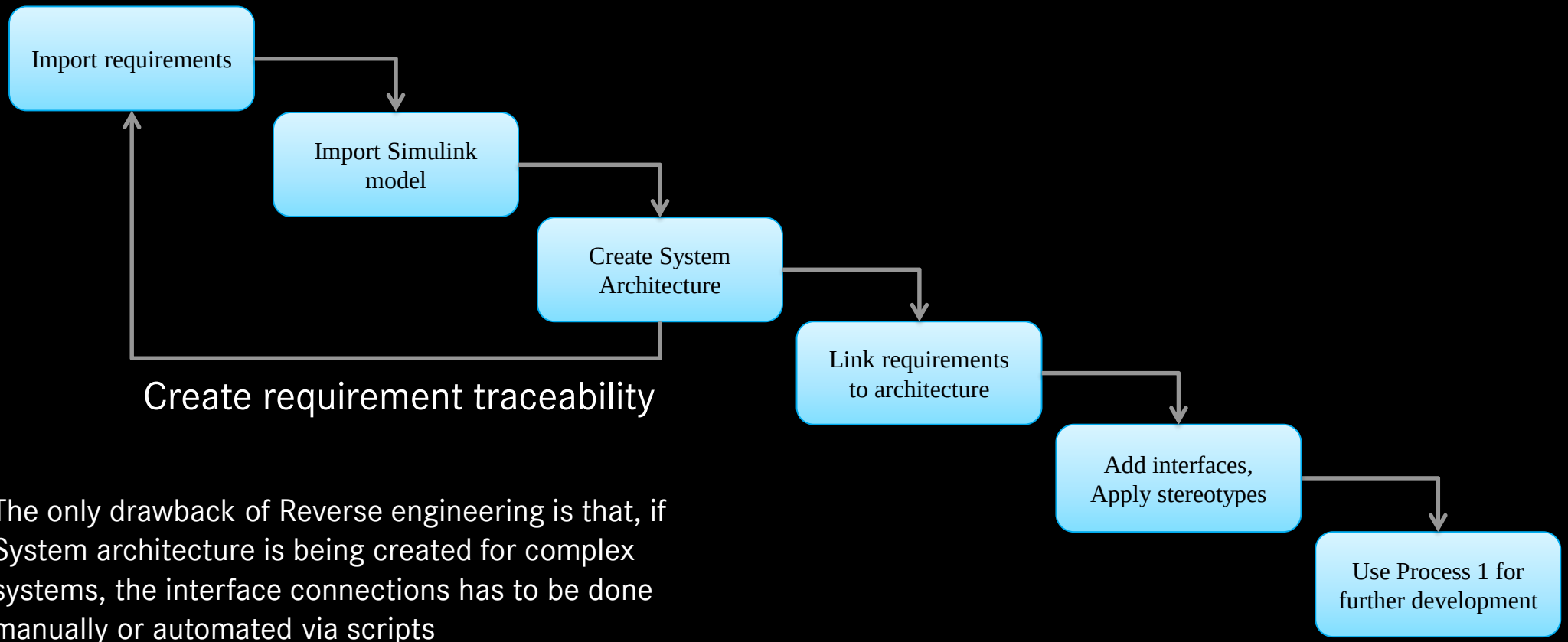


- In this process, the developer has access to the entire architecture while implementing the software and could understand how the requirements are affecting the entire system with functional background
- Easy to understand the functionality even if the topic is new to the developer

- If you use System composer just as a Requirement specification for Software development, the developer misses out on features like Writing test specifications against each requirement, seamless integration between System and software architecture
- One more additional tool to be incorporated in the software development process

Our findings after using System Composer

Process 1c: Software to System Architecture (Reverse engineering)



The only drawback of Reverse engineering is that, if System architecture is being created for complex systems, the interface connections has to be done manually or automated via scripts

Conclusion

- Communication between stakeholders and functional developers will be improved
- The tool is assistive in complex system development
- As it is a hierarchical modelling, what and where to implement is pretty clear
- This process facilitates impact analysis of requirements and design changes
- Improves design quality by reducing errors and ambiguity
- Early and on going verification and validation can be done to reduce the risk
- Enhances knowledge on the system

Questions?

Thank you