Integration of Requirements Management and Architectural Modeling

Kathy Culver
Applications Engineer
Telelogic, North America
kathy.culver@telelogic.com

• Not a presentation focusing on requirements or requirements management…

….but will touch on why requirements are important.

• Not a presentation focusing on architecture methods and notation…

….but will mention some of them by way of example.

This is a presentation on how combining architecture models with requirements can be effective for……

- Enhancing communication with customers, development team, and subcontractors, thereby reducing the chances of misinterpretation of data and concepts.

- Smoother integration of components and systems (SoSE)….fewer surprises.

- Verify that systems being built perform to specification

What are Requirements?

(They are the TO-DO List of the Project Team)

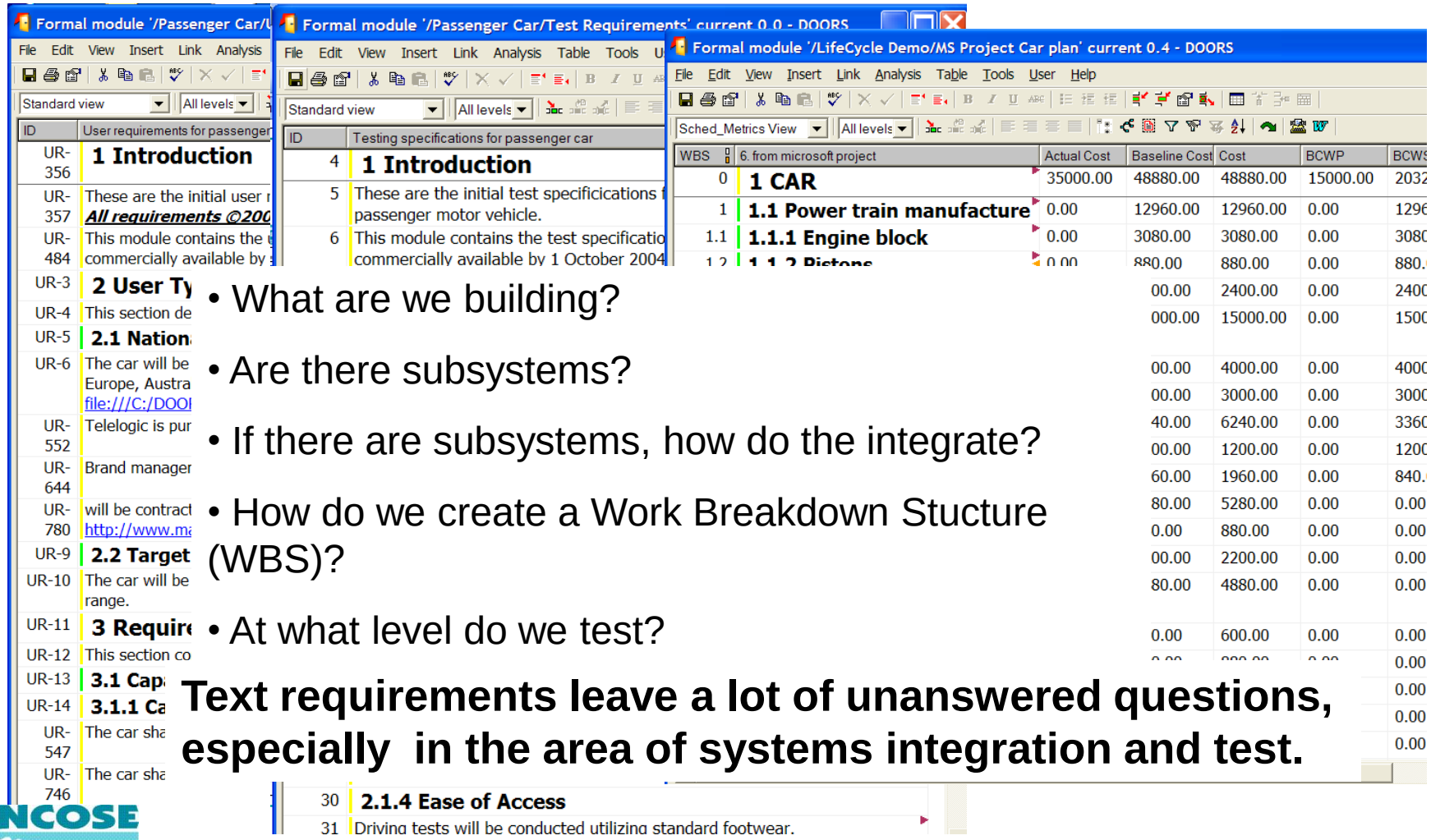
- List of the goals and objectives of the business
- List of what the users need
- List of what the system must do to satisfy user and business needs
- List of what components must be built
- List of what each component must do, and how components will interact

The Role of Requirements

- Come to an agreement with the customer and users on what the system should do
- Give system developers a better understanding of the system
- Delimit the system
- Provide basis for planning technical iterations
- Provide basis for performing system tests (Verification)
- Provide a basis for acceptance (Validation)

Are textual requirements enough…..

….to effectively and efficiently build, integrate and deploy a system or System of Systems?



- What are we building?
- Are there subsystems?
- If there are subsystems, how do the integrate?
- How do we create a Work Breakdown Stucture (WBS)?
- At what level do we test?

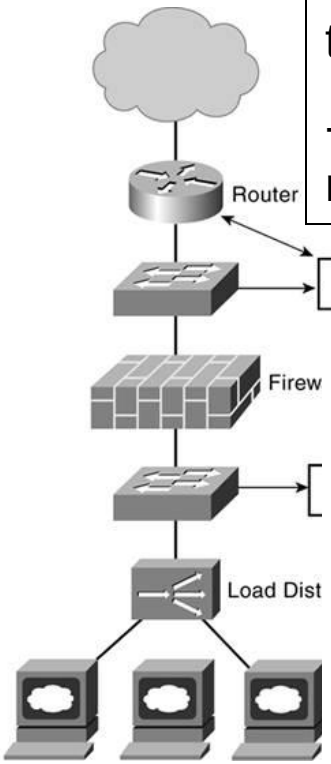
Text requirements leave a lot of unanswered questions, especially in the area of systems integration and test.

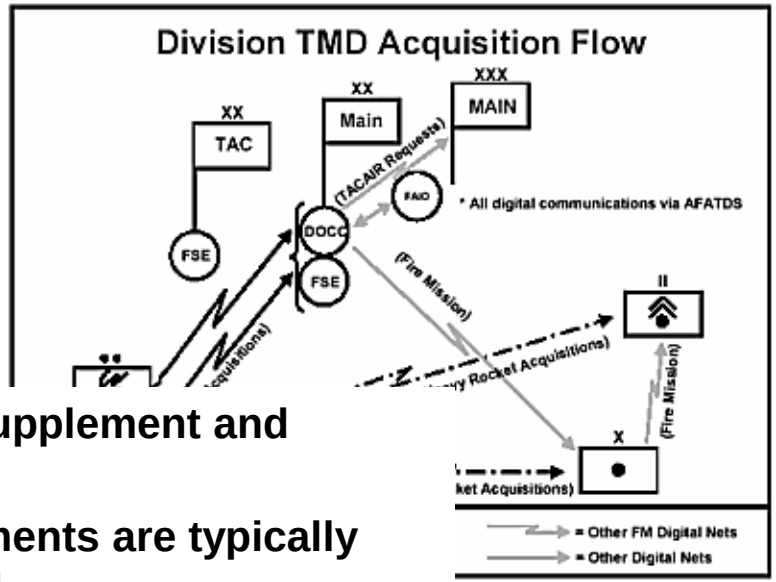
INCOSE San Diego CHAPTER

The Model is ***not*** the Requirement

- What are the goals of the system?

- What are the user needs?



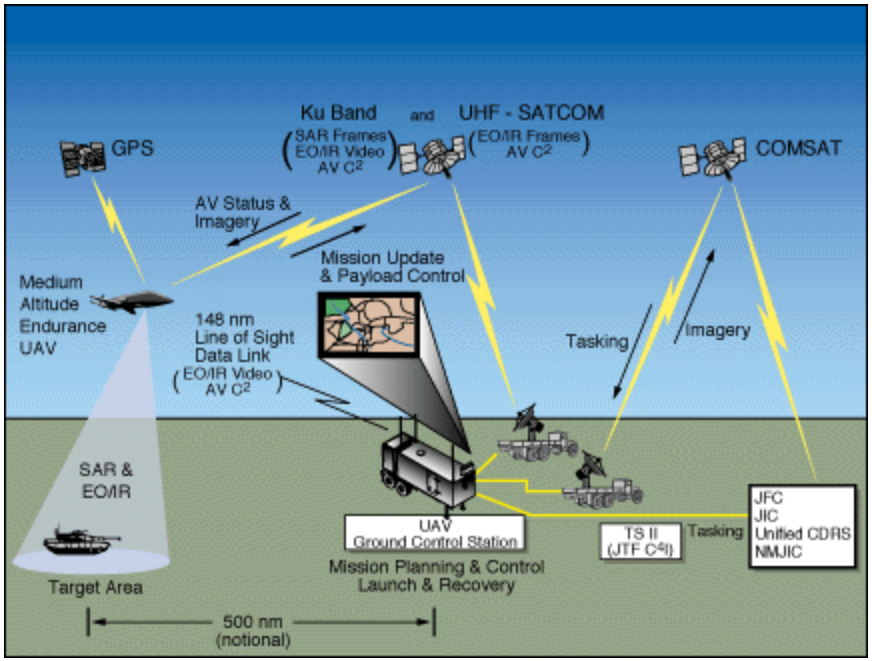


- **Textual requirements supplement and explain the models**
- **non-functional requirements are typically not captured in a model**
  - **Performance**
  - **Safety**
  - **Ease-of-use**
  - **Time lines**
  - **Etc…**
- **a graphical model is generally insufficient as a contractual basis.**

INCOSE San Diego CHAPTER

Now we can see the Big picture…

Requirements Document



- We know what we are building.
- There are subsystems.
- We understand high level integration.
- Rough idea of Work Breakdown Structure (WBS).
- Rough idea of test.

INCOSE San Diego CHAPTER

Managing Complexity – Divide and Conquer

Relating Requirements
To
Systems of Systems Engineering (SoSE),
Systems Engineering

INCOSE San Diego CHAPTER

Capability Driven, Architecture Centric, Model Based Club Sandwich

Needs (problem)

Modelling layer

Capability (problem/solution)

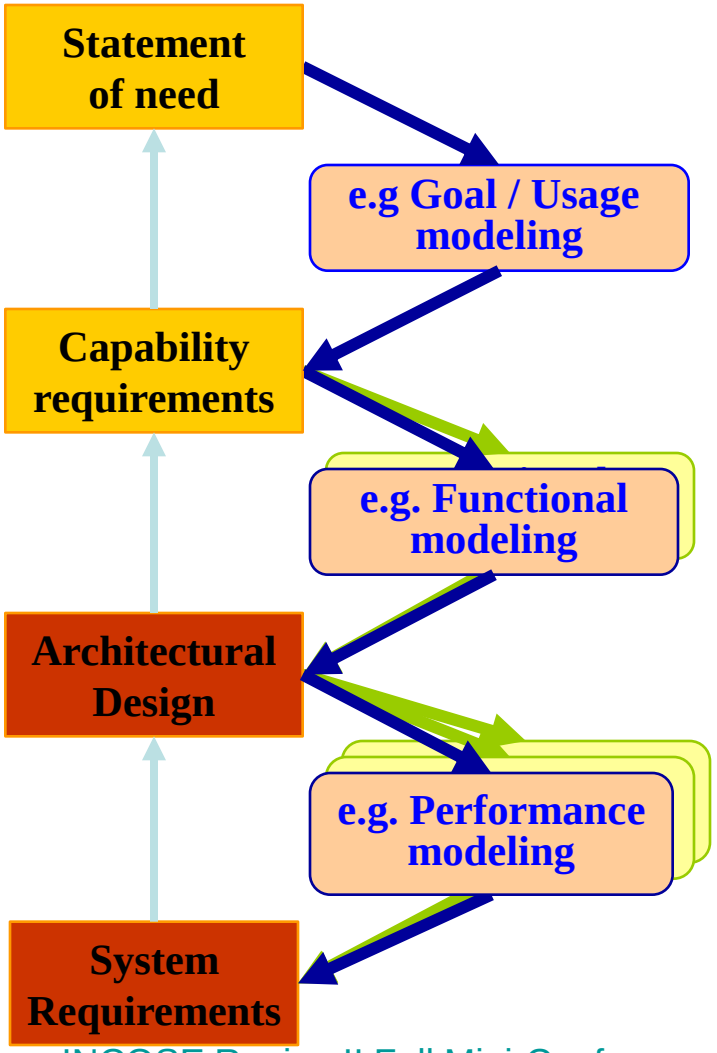
Modelling layer

Requirements (solution)

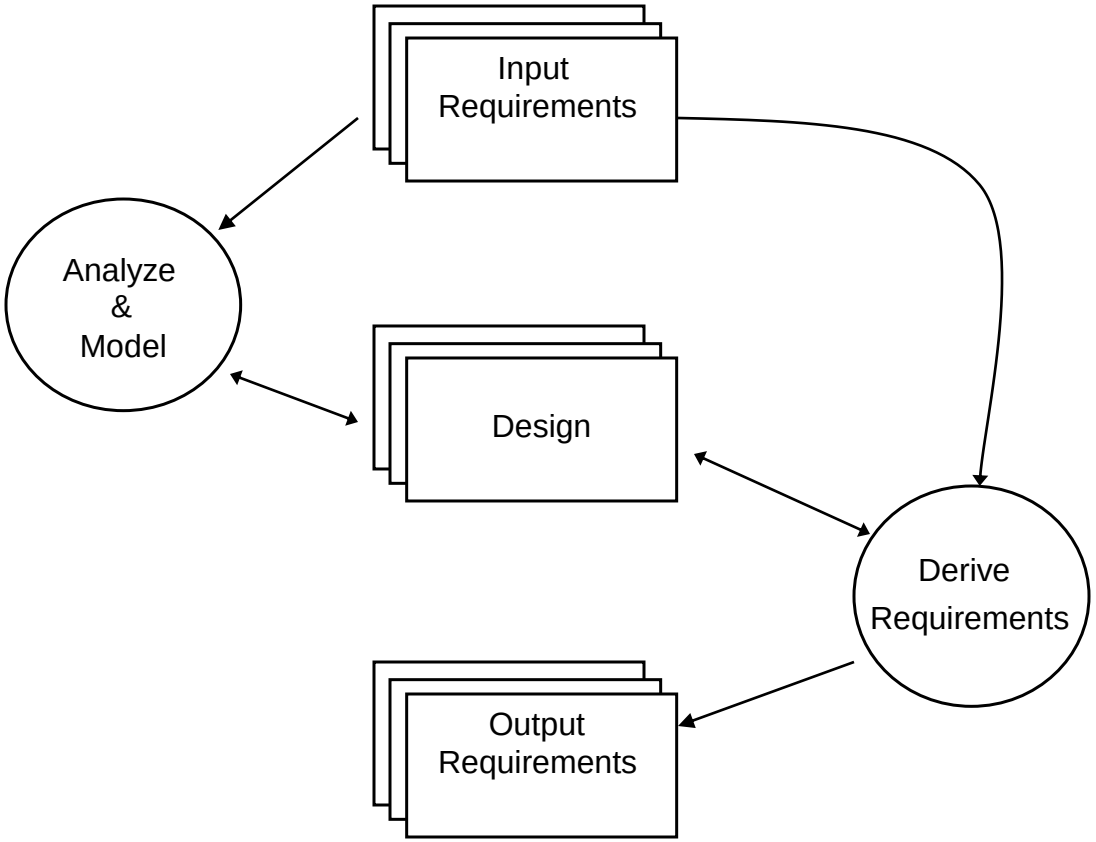
Modelling layer

Requirements (solution)

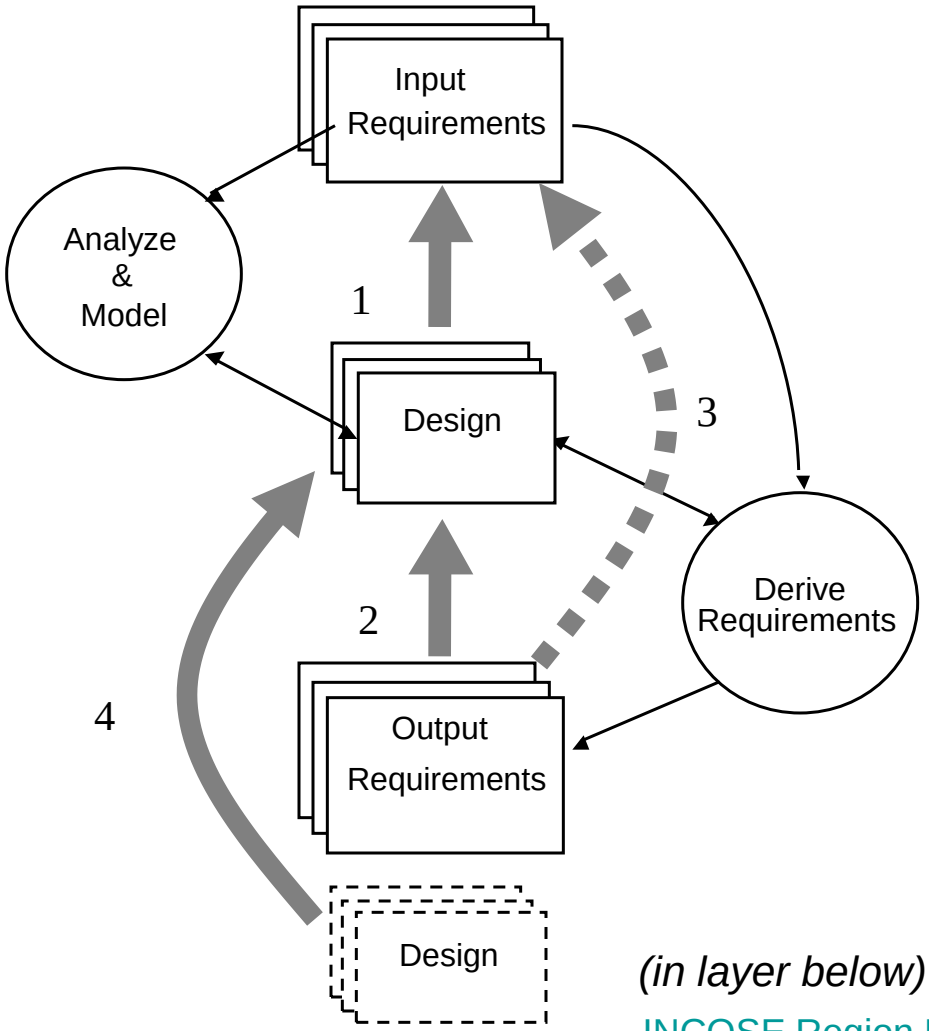
Models Bridge Layers of Requirements



Basic Process for Systems Engineering

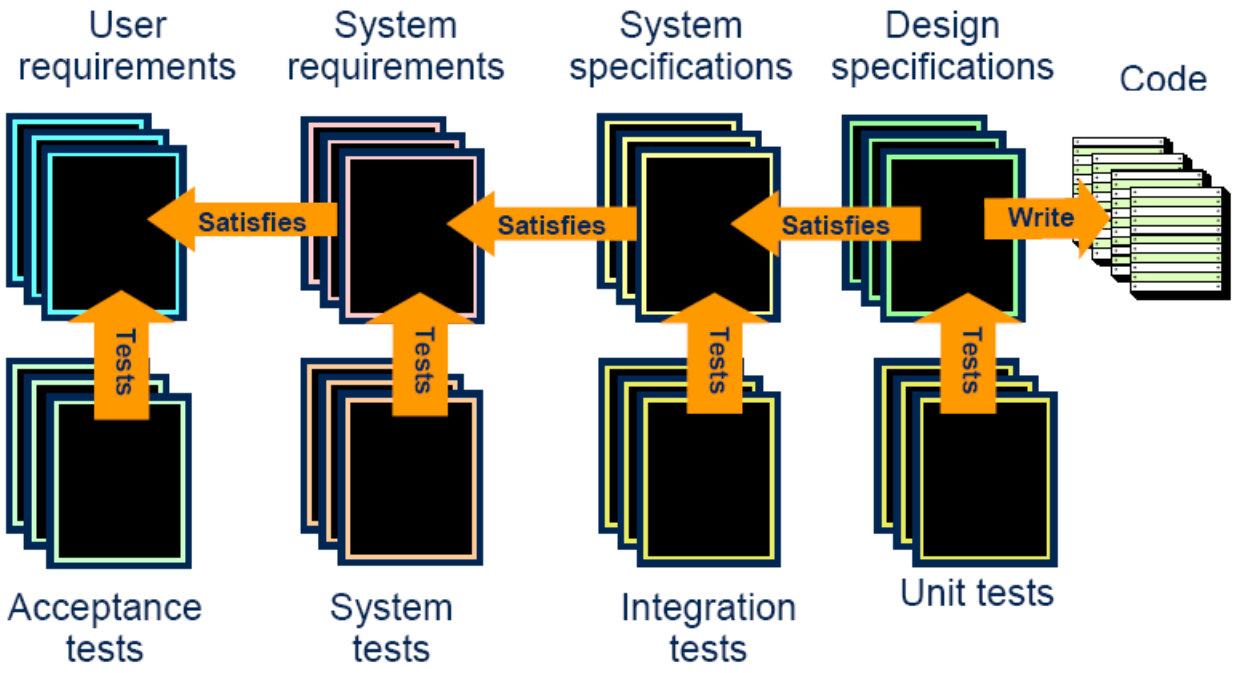


Basic Process for Systems Engineering Showing Traceability

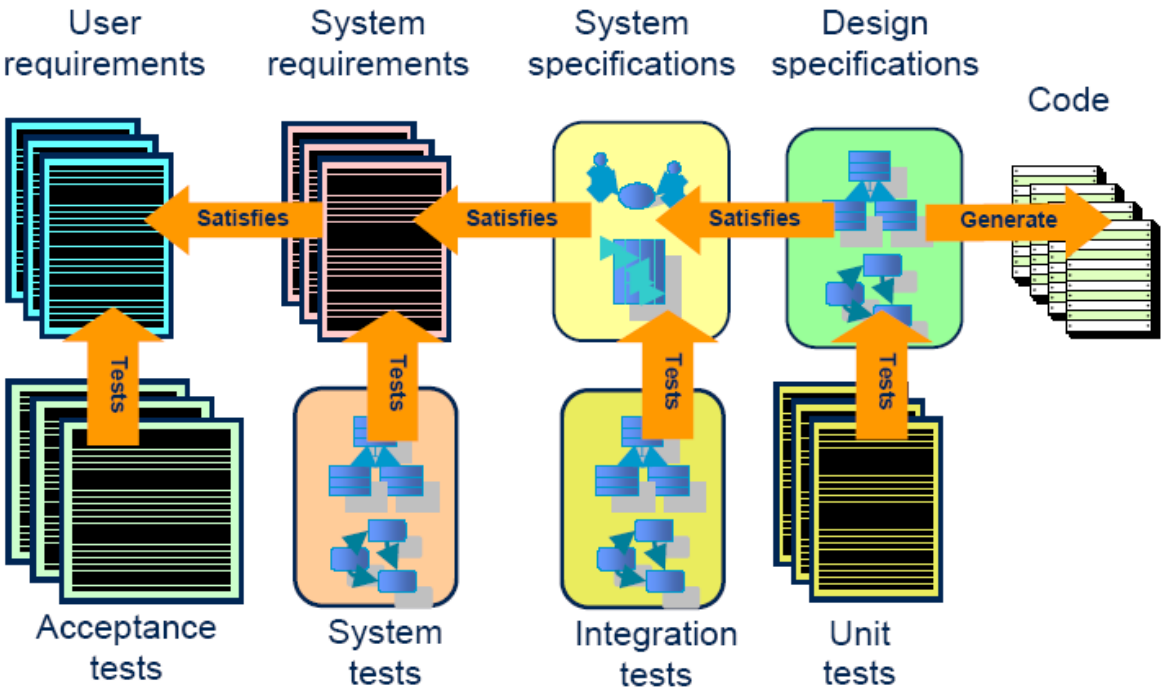


(in layer below)

In traditional requirements management, documents are produced, and relationships between elements of those documents are established, as outlined below:



Modeling has been shown to be an essential part of project development, aiding in the visualization and clarification of requirements and assuring their robustness and structural integrity.



A natural flow is established from those setting the original requirements to those developing and launching the final product,

Integrating Requirements Management and Architectural Modeling

Examples:

Department of **D**efense **A**rchitectural **F**ramework - **(DoDAF)**

System **M**odeling **L**anguage – **SysML**

Simulation for Requirements Verification

What is DoDAF (Department of Defense Architecture Framework)?

- ***"The DoDAF version 1.0 defines a common approach for DoD architecture description, development, presentation and integration for both warfighting operations and business processes. The DoDAF is intended to ensure that architecture descriptions can be compared and related across organizational and mission area boundaries, including joint multi-national boundaries and DoD warfighting and business domains."***
  - Excerpt from memo from John P. Stenbit, CIO, Department of Defense, February 2004.

- DoDAF supersedes C4ISR Architecture Framework

Interoperability Is Key To Successful Military Operations

- Breakdown in communications leads to:
  - 'Friendly fire' incidents
  - Lack of co-ordination of units
- 'Net-Centric Operations and Warfare' is the solution
  - Effective communications between forces
  - Compatible technologies
  - Interoperable systems
- Requires a standard way to describe systems and their interfaces
  - So that 'touch points' can be checked for compatibility before the system is developed
  - Helps when new capabilities are 'grafted' onto existing systems

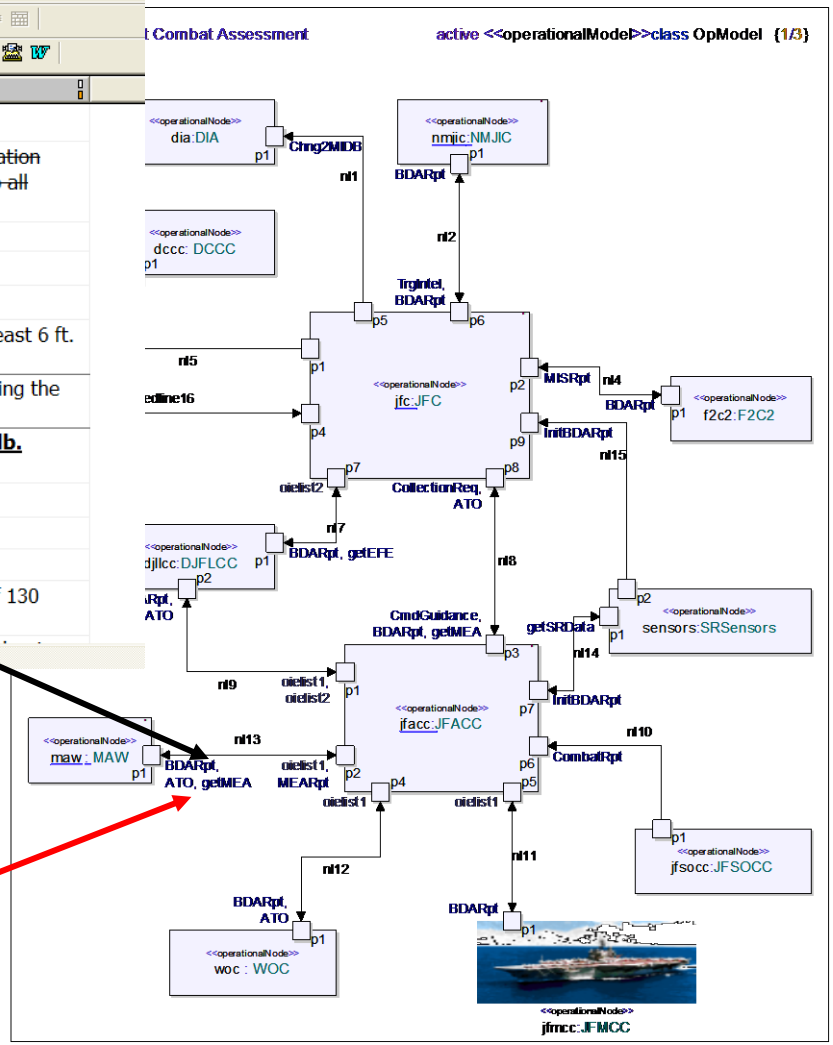
DodAF – OV-2 Operational Node Connectivity

Formal module '/Sample/MFCS' current 0.0 - DOORS

File Edit View Insert Link Analysis Table Tools User Help

Standard view | All levels

ID	Missile Fire Control System
req-1	**1 System Interface definitions**
req-2	~~The system shall receive radar messages containing Target Tracking information (Position, Heading, Height, Speed, Identification Friend or Foe (IFF) code fro all tracks from the~~ radar system.
req-3	The system shall assign a track ID to each track. a change
req-24	**1.1 *Signal Definitions***
req-25	**1.1.1 Subsystem Interfaces**
req-26	• Driving, safety and road tests will be conducted by a test driver at least 6 ft. 5 in. in height.
req-27	• Road test will be conducted utilizing four average size adults satisfying the level of comfort provided by the top 40% of cars produced in 2000.
req-28	• Driving, safety and road tests will **be conducted utilizing a 240 lb. weight in the luggage** compartment.
req-29	**1.1.2 Signal Field Defintion**
req-30	**1.1.2.1 Speed & Acceleration**
req-31	**1.1.2.1.1 Signal Timing specs**
req-32	A radar gun will be utilized to determine the vehicle can maintain a speed of 130 mph on a flat, straight road with minimal wind conditions.



OIEs – show interfaces between operational nodes

Can be linked to

Interface Description Reqs

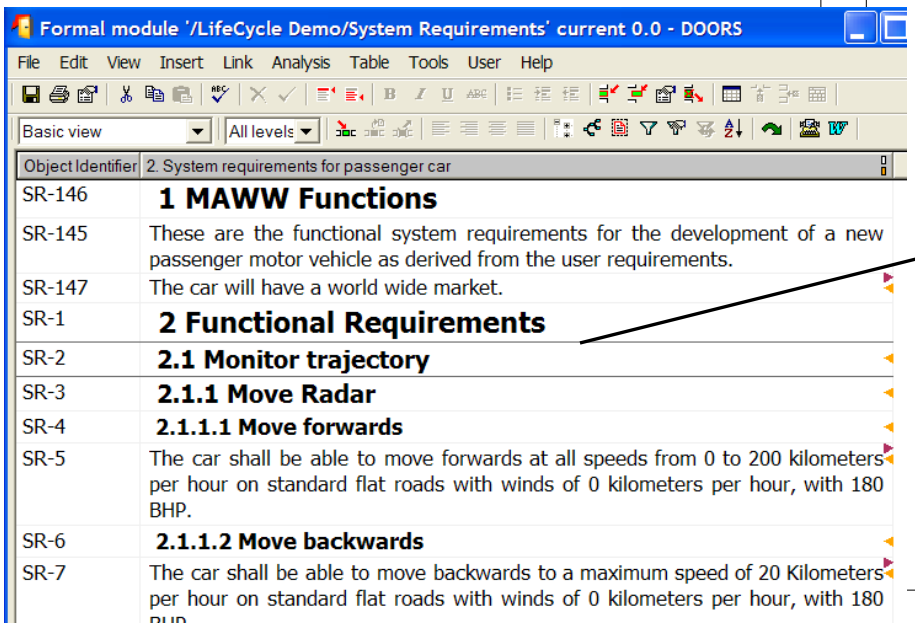
Information Exchanges

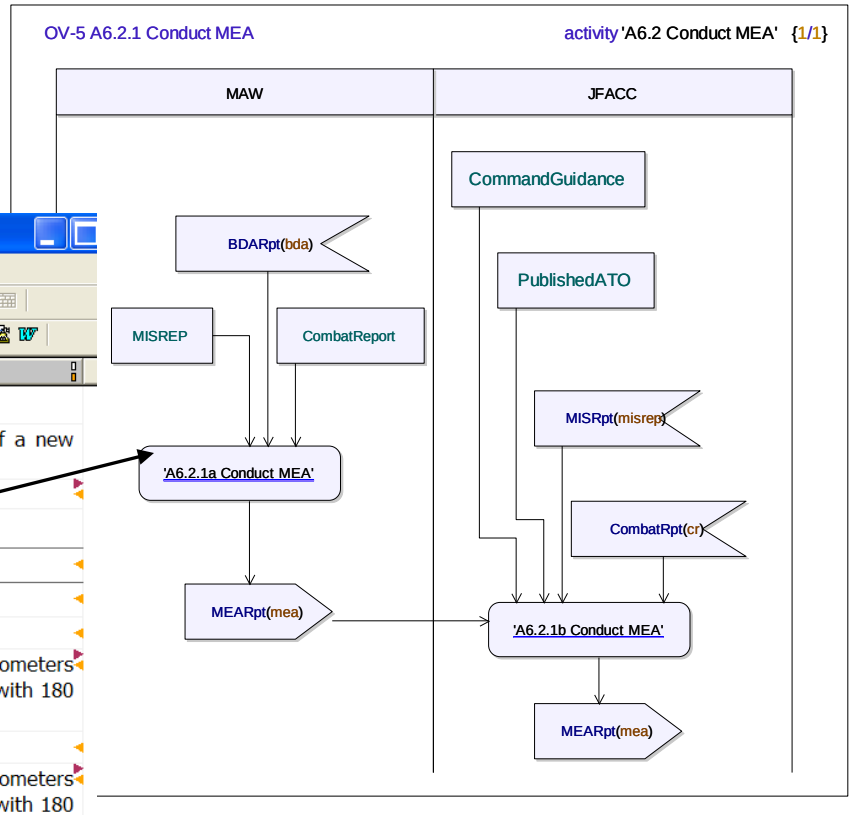
DodAF – OV-5 Operational Activity

OV-5 decomposition of activity per Op_Node

links to

Functional Requirements

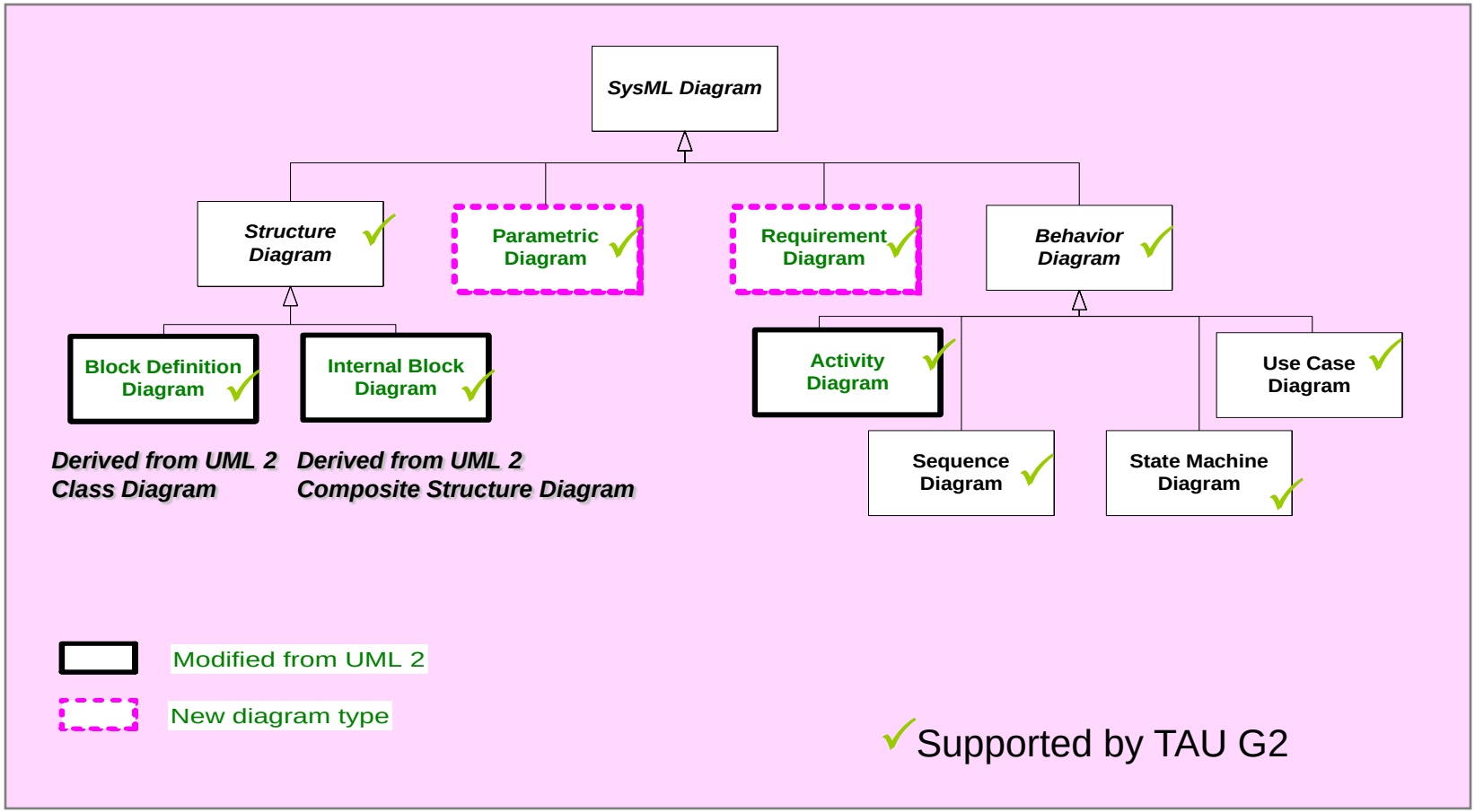




What is SysML (System Modeling Language)?

- **Systems Modeling Language** (**SysML**) - an extension of the UML for systems engineering applications. SysML supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems. These systems may include hardware, software, information, processes, personnel, and facilities.
  - SysML is an open source project that is organized and supported by representatives from the SysML Partners, an informal association of industry leaders, tool vendors, government agencies and professional organizations.

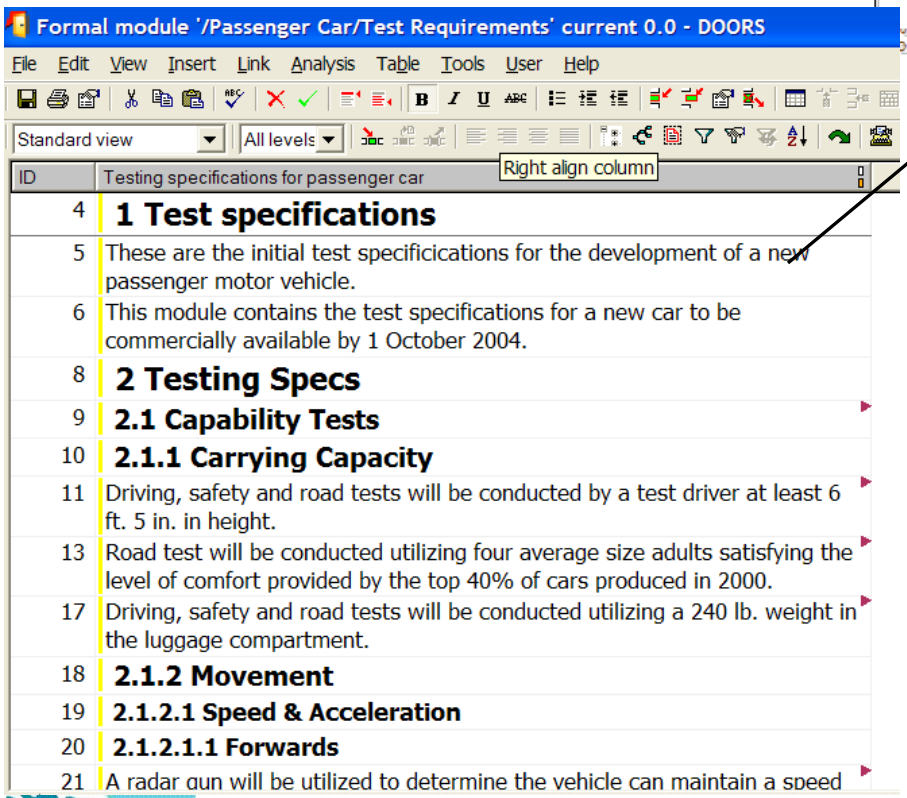
SysML Diagram Taxonomy

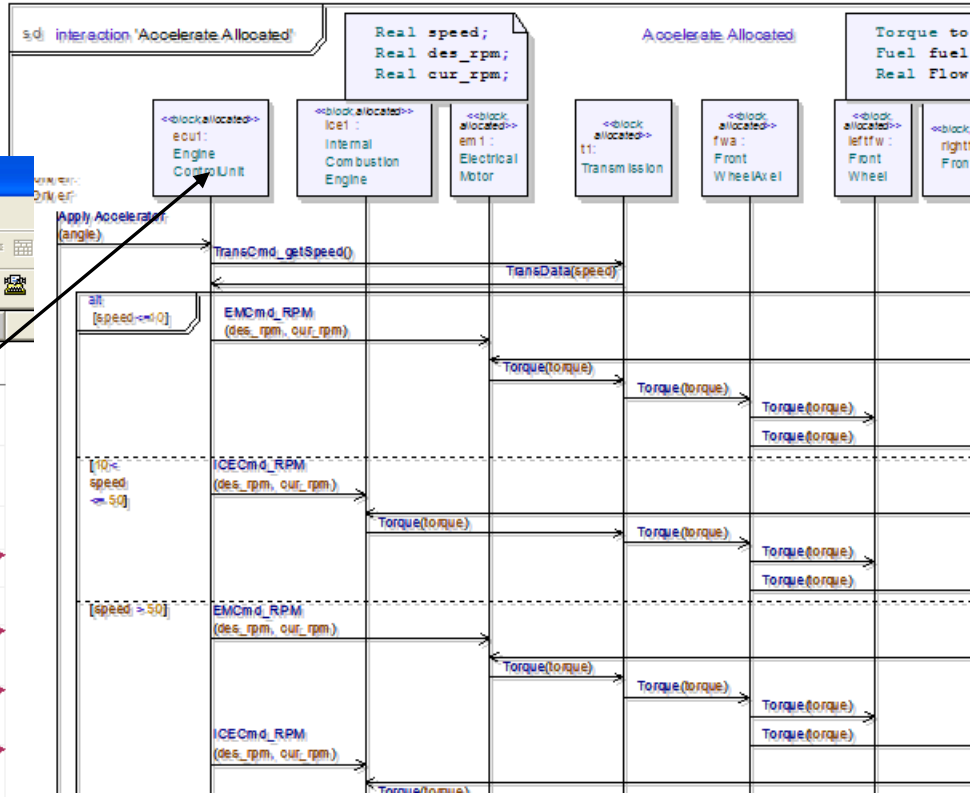


SysML – Sequence Diagram

- Shows control and data flow
- Useful for analyzing key system scenarios and response threads.

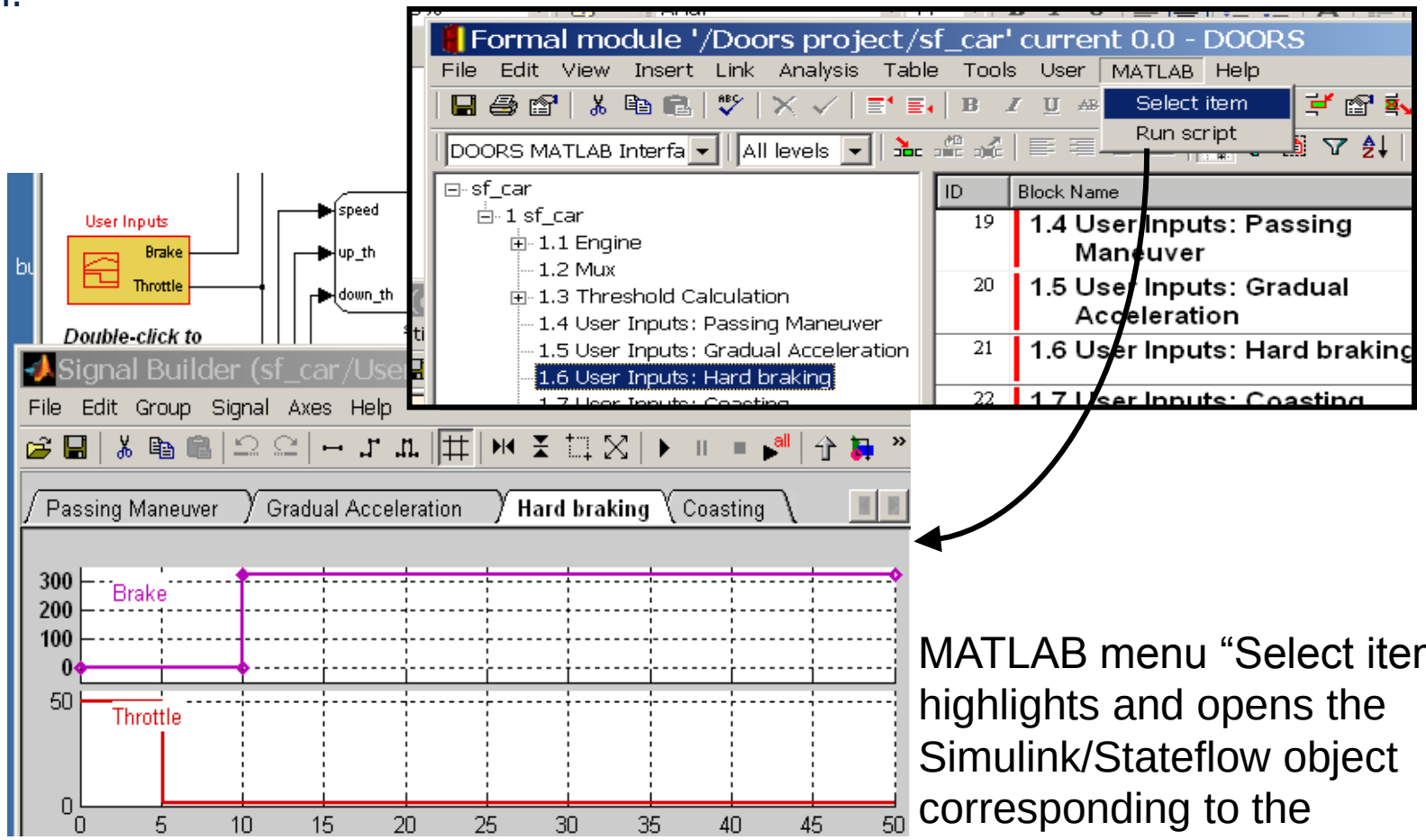
Can be linked to
Test specifications to verify sequence





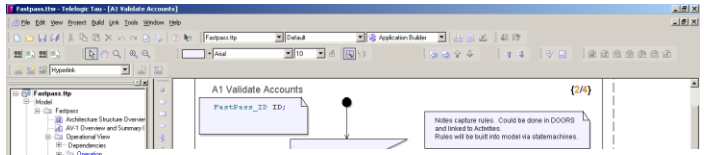
Requirements Verification and Validation using MatLab for Algorithmic Simulation

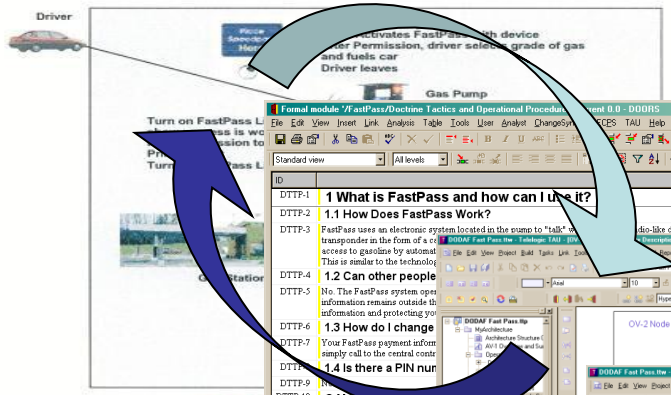
- MATLAB is well suited for complex algorithm development. The elements derived from the MathWorks suite of tools are linked back to the requirements as well.

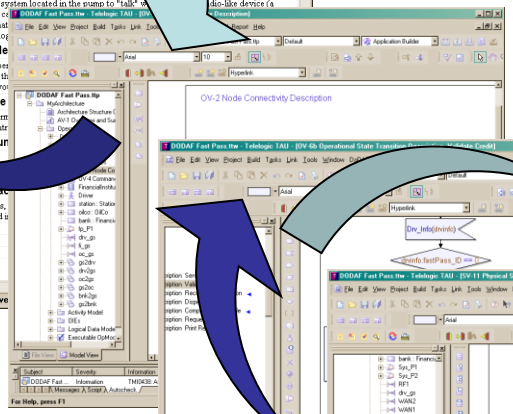


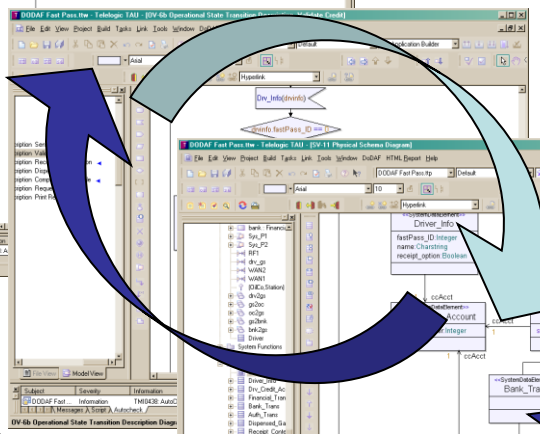
MATLAB menu "Select item" highlights and opens the Simulink/Stateflow object corresponding to the selected row.

Integrate Thoughout the Lifecycle









Reuse

Business Process

A bright idea!

Requirements Development

Requirements Analysis

Systems Architecture

Systems Modeling

Software Design

Application

Traceability/

Verification

INCOSE San Diego CHAPTER

Tool Support for Integration of Requirements and Architecture Models

Telelogic – DOORS, System Architect, Tau, Rhapsody (fully integrated)

IBM/Rational – Requisite Pro, Rose, RSA

UGS - SLATE, Teamcenter for Requirements

Others – Visio, Excel, Word…"roll your own" etc.

Summary:

- Text requirements can leave a lot of unanswered questions, especially in the area of systems integration and test.

- The Model is *not* the Requirement

Benefits of an integrated approach:

- Aids communication with customers, development team, and subcontractors, thereby reducing the chances of misinterpretation of data and concepts.
- Smoother integration of components and systems (SoSE)…..fewer surprises.
- Requirements validation and verification can be achieved through links to simulation in the modeling environment.

Questions