Functional Specifications … What for?

One of my mottos could be:

“Make things simple & easy!”

But we work in a world of Huge Complexity.

“Simplicity is the ultimate sophistication”

Leonardo Da Vinci
Functional Specifications … What for?

One of my mottos could be:

“Make things simple & easy!”

“True simplicity is derived from so much more than just the absence of clutter and ornamentation. It’s about bringing order to complexity...

You have to deeply understand the essence of a product in order to be able to get rid of the parts that are not essential.”

Jony Ive  (Chief Design Officer at Apple)
How to address Complexity?

➔ Break complexity into sets of simpler parts.
How to address Complexity?

➔ Break complexity into sets of simpler parts.
➔ Design each elementary part …
➔ manage integration.

How to ensure that this giant puzzle will answers global need?
Through **Model based design**, using
- Simulation,
- Prototyping,
- Software In the Loop,
- Hardware In the Loop.

Early Testing & Corrective loops

How to ensure that this giant puzzle will answer global need?

Through **System Engineering** and **Requirements cascading**... using, **Model based System Engineering**
Functional Specifications … What for?

How to capture and specify user needs, at any level?
How to capture and specify user needs, at any level?

Capture intended behaviour, telling stories of use.

- **Natural way** to formalise expected behaviour.
- At the **right level** (without constraining to a foreseen design solution)
- Allows focusing on **some simple cases**, for some given conditions ➔ no need to formalise all potential cases

But each scenarios is only a part of the global image!
How to capture and specify user needs, at any level?

1. **Stakeholder Functional Requirement**
2. **Global functional description**
   - Build global functional descriptions from scenarios.
3. **Scenarios of use**
   - Provide the **global image**, to support **requirements identification** and specification building.
How to capture and specify user needs, at any level?

Let’s consider a simple example of interactive User Interface:

- Control command using the mouse to interact with a slider, with 3 types of interactions:
  - Drag & drop of the cursor widget
  - Using the mouse wheel
  - Clicking on + / - button

![Diagram of interactive User Interface with a slider and buttons](imaginary-image-url)
How to capture and specify user needs, at any level?

Scenarios of use

Model actions done by user.

Model associated functions.

Model functions’ triggers & data exchanges.

Global functional description

Stakeholder Functional Requirement

* Customizations developed by MathWorks based on MATLAB®, Simulink® & SimEvent®.
How to capture and specify user needs, at any level?

Scenarios of use

Global functional description

Stakeholder Functional Requirement

Automatically build a Simulink model that merges scenarios.

*Customizations developed by MathWorks based on MATLAB®, Simulink® & SimEvent®.*
How to capture and specify user needs, at any level?

- Stakeholder
- Functional Requirement
- Global functional description
- Scenarios of use

Automatically build a textual specification.

*Customizations developed by MathWorks based on MATLAB®, Simulink© & Simulink Verification & Validation©.
An Efficient Scenarios-Based Solution to Ensure Completeness and Consistency of Functional Specifications.

How to capture and specify user needs, at any level?

Scenarios of use

Global functional description

Stakeholder Functional Requirement

How to ensure that final specification is correct & complete?

*Customizations developed by MathWorks based on MATLAB®, Simulink® & Simulink Verification & Validation®.
How to capture and specify user needs, at any level?

- Scenarios of use
  - Modifications
  - Simulation
  - Executable model

Customizations developed by MathWorks based on MATLAB®, Simulink® & Simulink Design Verifier®.
How to capture and specify user needs, at any level?

- Stakeholder Functional Requirement
- Global functional description
- Scenarios of use
- Modifications
- Simulation
- Executable model
- Model functional constraints
- Formal proof

*Customizations developed by MathWorks based on MATLAB®, Simulink® & Simulink Design Verifier®.
Conclusion

✅ **Scenarios** modelling:
- Easy & natural way of representation (simple *user stories*)
- Stops decomposition at the right level
- Data formalization & data capitalization (share, re-use…)

✅ **Automatic** generation of functional architecture:
- Auto-structures the specification
- Auto-formalizes part of the requirements

✅ **V&V** improved:
- Validation through enlarged *simulation* capabilities
- Formal verification of behavior against requirements

Let’s make this Proof of Concept, an **Industrial reality**.
An Efficient Scenarios-Based Solution to Ensure Completeness and Consistency of Functional Specifications.

Q&A