



MBSYS-02
Practical Model Based Systems Engineering with SysML
notation and SysML tool – 4 Days
Agenda

contact@samares-engineering.com

- Learn practical MBSE approach from requirements to architecture down to execution platform
 - Based ISO 15288:2015 technical processes
 - Using SysML notation for system global definition (requirements and architecture)



- Case study for practice
 - UAV for agriculture as System of Interest
 - Initial requirements from DOORS or Excel
 - Exercises on case study
 - Use of tool (Cameo Systems Modeler or other)



Requisites for attendees

- First knowledge on Requirement Engineering
 - Good quality: Single goal, affordable, verifiable, ...
 - « System shall... »



- Course introduction
 - Quick presentation of attendees and trainer
 - Presentation of case study
- Capture system physical boundaries
 - Identify target links with IBD
- Capture system functional boundaries
 - Capture of functional requirements with Use Case Diagram (UCD)
- Capture of scenarios with time boundaries and traceability
 - Define scenarios with Sequence Diagram (SD)
- Identification of system top-level functions and functional flows
 - Derive functions and functional flows from SD
- Capture other requirements as text
 - Use Requirement Diagram and Requirement table to complete StReq



- Recall of Day 1 activities
- Complete top-level functions definition
 - Use AD to complete functional definition (control and functional flows)
 - Use SMD to formalize modes and states
 - Use Matrix to relate functions to modes and states
- Constraints – binding of key parameters
 - IBD to formalize constraints on connected systems
- Design of lower-level functions
 - Use AD to refine high level functions into lower level functions
 - Ensure data continuity with functional flows



- Recall of Day 2 activities
- Functional trade-off
 - Different designs for a given function
- Complete system requirements
 - Use Requirement table to complete formalized SysReq
- Ensure traceability to StReq
 - Define Requirements Traceability Matrix
- Formalize System elements and logical Architecture
 - Use of BDD for breakdown structure



- Recall of Day 3 activities
- Definition of complete logical system architecture
 - Use of Allocation matrix to map functions to logical elements
 - Distribution of functional flows on logical architecture (IBD)
- Comparison and selection of “optimal” architecture with Par
 - Detailed equations for cost and performance criteria
- Detailed design of each logical System Element
 - Subsystem development versus acquisition
- Refinement of logical element into physical solution
 - BDD with inheritance and instance table to present design options
- Summary

