



DANSE Methodology Training

Target Audience

- Qualified systems engineers who are working on Systems of Systems
- Technology providers who want more information on SoS tools
- Researchers advancing the state of the Systems Engineering art

Course Duration

- 2 sessions of 2 hours each – 4 hours of instruction
- Delivered via webinar

Instructional Methods

Method	Target
Lecture	60%
Discussion	20%
Short Exercises (instructor-led)	20%
Major Exercise	None

Topical Outline

TOPIC		DURATION
Lesson 1	Course Introduction	Day 1
Introduction <ul style="list-style-type: none"> • Purpose of training • Expected goals for the training 		0:15
DANSE Systems of Systems Concepts <ul style="list-style-type: none"> • What is the nature of an SoS for which this methodology applies • SoS creation and management • Why is the methodology necessary? • Dynamicity and emergent behaviour 		0:15
Lesson 2	DANSE Methodology Overview	Day 1
DANSE Methodology <ul style="list-style-type: none"> • Two paradigms of SoS creation/management <ul style="list-style-type: none"> • Pre-existing SoS to be evolved adaptively • New SoS to be created and evolved • Five primary tasks in the methodology <ul style="list-style-type: none"> • Model SoS behavior • Operate the SoS • Define potential needs • Analyze possible architectural changes • Influence and implement changes • The Capability Learning Cycle - Iteration in the methodology • Relationship of the SoS evolution to constituent systems 		0:15
DANSE Solution Methods <ul style="list-style-type: none"> • Twelve solution methods available to the SoS manager/architect. 		0:10

TOPIC		DURATION
<ul style="list-style-type: none"> How the solution methods support the engineering activities in the SoS life cycle. When to use them, and what to expect. The tools that support each solution method. Exercise: Airport SoS 		
DANSE Tools <ul style="list-style-type: none"> List of DANSE tools The tools that support each solution method. 		0:05
Lesson 3	Model SoS Behavior	Day 1
SoS Definition <ul style="list-style-type: none"> Understanding the SoS, bounds, goals, modeling goals Characteristics of interest Global and local goals and their conflicts 		0:10
SoS Modeling <ul style="list-style-type: none"> Solution method 1: Model SoS, and its tools General models – UPDM Desired joint simulation capabilities Exercise: Emergency response SoS 		0:10
Constituent Systems Modeling <ul style="list-style-type: none"> Situated models – characteristics of interest Typical types of models 		0:10
Abstraction Methods <ul style="list-style-type: none"> Solution method: Abstract CS model Fitting constituent systems models into the SoS models Three abstraction methods Exercise: Emergency response SoS 		0:10
Tool-Net Capabilities <ul style="list-style-type: none"> Joint tool use concepts Tool-net connections and structure Example: Include SysML structure within UPDM Solution method 12: Share Models Solution method 11: Configure DANSE Tool-Net environment 		0:10
Lesson 4	Operate the SoS	Day 1
SoS Capability Management <ul style="list-style-type: none"> Monitor the characteristics of interest Observe behaviors in operation 		0:10
End of Day 1		
Lesson 5	Define Potential Needs	Day 2
Adaptive Evolution <ul style="list-style-type: none"> Evolution concepts Extracting evolutionary needs from the observations 		0:10
Goals and Contract Specification Language (GCSL) <ul style="list-style-type: none"> Semantically sound constraints that can be automated GCSL syntax and GCSL Editor Solution method 9: Evaluate goals and contracts 		0:10
Lesson 6	Analyze Possible Architecture Changes	Day 2
Architecture Patterns <ul style="list-style-type: none"> Concepts of patterns Pattern language – how to convey patterns to others Pattern usage in DANSE Solution method 3: Apply architecture patterns 		0:15
Architecture Generation: Graph Grammar <ul style="list-style-type: none"> SoS models as graphs Graph transformations to generate new architectures 		0:15

TOPIC	DURATION
<ul style="list-style-type: none"> • Reachability of future architectures • Example: Emergency Services SoS • Solution method 4: Generate architecture alternatives 	
Architecture Optimization: Concise Modeling <ul style="list-style-type: none"> • Concise modeling as tabular enhancements to SysML • Optimization and multi-objective optimization • Dashboard for architecture optimization • Solution method 5: Generate optimized architectures 	0:15
Joint Simulation: Performance Evaluation <ul style="list-style-type: none"> • Joint simulation capabilities and tools • Use of joint simulation to check SoS performance • Solution method 6: Perform joint simulation • Solution method 10: Perform formal verification 	0:15
Statistical Model Checking <ul style="list-style-type: none"> • Statistical model checking concepts • Checking architecture alternatives against GCSL • Solution method 7: Perform statistical model checking 	0:10
Emergent Behavior <ul style="list-style-type: none"> • Concepts of emergent behaviours • Classical solutions for emergence • Four DANSE methods to discover/predict emergence • Solution method 8: Evaluate emergent behaviour 	0:10
Lesson 7 Influence and Implement Changes	Day 2
Changing the SoS <ul style="list-style-type: none"> • System requirements derived from SoS analysis • SoS control versus SoS influence • Implementing SoS changes by modifying the CSs 	0:10
Lesson 8 Course Summary	Day 2
Course topics review <ul style="list-style-type: none"> • Selected slides repeated from course Course evaluations Course certificates	0:10