

1. Tutorial – max 45 Teilnehmer

Modellierung operationeller Architekturen im MBSE mittels dem NATO Architecture Framework (NAF)

Dr.-Ing. Martin Becker, Markus Marx, Oliver Bleisinger (Fraunhofer IESE)

Voraussetzungen: ggf. wird ein eigener Laptop benötigt

Dieses Tutorial soll eine kurze Einführung in die Modellierung operationeller Architekturen basierend auf dem NATO Architecture Framework v3.1 liefern. Der Fokus liegt hierbei auf der Modellierung verschiedener Sichten der operationellen Architektur und dem Aufzeigen des Mehrwerts für die Arbeit in interdisziplinären Projekten bzw. Projektteams. Ein besonderer Fokus liegt auf der operationellen Architektur.

2. Tutorial – max 15 Teilnehmer

Testgetriebene und testunterstützte Modellierung

Rupert Schlick (AIT Austrian Institute of Technology)

Voraussetzungen: Basiskenntnisse UML/SysML sind hilfreich

Speziell wenn Modelle für die Generierung von Code für Steuerungssysteme verwendet werden sollen, stellt sich die Frage, ob denn das Modell „richtig“ ist. Für Domänenexperten sind Modelle zwar bereits zugänglicher als Programm-Code, die Frage, ob ein Verhaltens-Modell tut, was es soll ist aber trotzdem nicht einfach zu beantworten.

3. Tutorial – max 24 Teilnehmer

Are you ready for the Moon? – Agile and Data Driven Systems Engineering

Setting up a Moonbase Architecture in 3 Sprints

Marco Witzmann (Valispace), Stephan Finkel (3DSE), Sebastian Märkl (3DSE), Benedikt Dechamps (3DSE)

Voraussetzungen: Mitnahme eines W-Lan-fähigen Notebooks erforderlich

Entlang einer Fallstudie werden Sie in 3 Sprints die Entwicklung einer Mondbasis vorantreiben und Architekturentscheidungen datengetrieben treffen. In kleinen schlagkräftigen Teams werden Sie nach Systems Engineering Methoden das Konzept aufsetzen. Dazu gehören klassische Elemente wie Funktionsanalyse, Systemdekomposition, Architekturentwicklung und das Bewerten von Alternativen. Daneben werden Sie Ihr Produktdatenmodell aufbauen und für wichtige Architekturentscheidungen nutzen. Das Ganze ist eingebettet in ein agiles Vorgehen mit entsprechenden Retrospektiven. Iterativ werden Sie Ihre Mondbasis, vom Groben ins Feine, immer weiter detaillieren und an diesem Fallballspiel die Verbindung von klassischen und agilen SE Methoden im Zusammenspiel mit Data Driven SE (DDSE) kennenlernen.

4. Tutorial – max 30 Teilnehmer

Agile Modeling in Safety Critical Environments

Wie agile Methoden das Modellbasierte Systems Engineering ASPICE-konform unterstützen können

Dr. Konrad Wieland, Roman Bretz (LieberLieber)

Voraussetzungen: Wissen über MBSE von Vorteil, UML/SysML hilfreich

Wie passt nun Modellbasierte Entwicklung mit agilen Methoden zusammen? Ist das nicht ein Widerspruch? Und können agile MBSE-Methoden auch ASPICE-konform umgesetzt werden? Diese und noch mehr hochaktuelle Fragen aus der Industrie werden wir in einem Tutorial beleuchten und im Detail diskutieren. Praxisnahe Beispiele von OEMs sowie Zulieferern im Automotive Bereich untermauern die Lösungsansätze und zeigen auf, warum Modellierung im sicherheitskritischen Bereich immer mehr eine zentrale Rolle einnimmt.

5. Tutorial (max 30 Teilnehmer)

Konsistenz von digitalen Systemmodellen

Prof. Dr.-Ing. Jutta Abulawi (HAW Hamburg), Prof. Dr.-Ing. Marco Di Maio (TH Ingolstadt)

Voraussetzungen: Praktische Erfahrungen mit MBSE

Das Ziel dieses Tutorials ist es, die Bedeutung der Konsistenz von Systemmodellen zu klären, Inkonsistenzen zu klassifizieren und Ursachen für ihre Entstehung zu identifizieren. Hierauf aufbauend werden Strategien zur Konsistenzprüfung und -sicherung erarbeitet.

6. Tutorial – max 14 Teilnehmer

Agile Hardwareentwicklung erleben mit Lego Mindstorms®

Alexander Holike, Bastian Fersch (HOOD)

Voraussetzungen: Die Teilnehmer brauchen keinerlei technische oder methodische Vorkenntnisse

7. Tutorial – max 30 Teilnehmer

Simulation and Verification of Cyber-physical Systems

Univ-Prof. Dr. Hermann Kaindl (TU Wien)

Voraussetzungen: EIN TUTORIAL FÜR FORTGESCHRITTENE UND EXPERTEN – SPRACHE IST DEUTSCH

This tutorial presents our investigation of (safety-critical) feature interactions and their coordination in automotive systems using traditional simulation, and our corresponding semi-symbolic AAF model of a CPS and its simulation. For formal verification, this tutorial shows minimalist qualitative models for *model checking* (safety-critical) CPS feature coordination and, that a specification of a software coordinator can be formally verified using the *Fluent Calculus* (a derivative of the *Situation Calculus*), when combined with additional models.

Based on more recent research, this tutorial also presents how to consistently and systematically derive a more concrete (quantitative) model from such an abstract (qualitative) model with regard to verification of its behavior against certain properties.

Stand 9.8.2019
Tutorial Beschreibungen