Dieses Wahlpflichtmodul ist ein Angebot der:

Fachhochschule
Dortmund
University of Applied Sciences and Arts

Master Embedded Systems for Mechatronics

Embedded Software Engineering

masteresm@fh-dortmund.de
+49 (0)231 9112-7991

Prof. Dr. Stefan Henkler
stefan.henkler@fh-dortmund.de

Embedded Software Engineering (MOD1-03)

Code Number	Workload	Credits	Semester	Frequency	Duration
10130/31	180 h	6	Sem. 1	annually	1 Semester

1	Course Title	Contact hours	Self-Study	Planned Group Size
	Embedded Software Engineering	4 SWS / 60 h	120 h	25 students

2 Course Description

Embedded software engineering is a multidisciplinary approach for developing Solutions to complex engineering problems. The continuing increase in system complexity is demanding integrated engineering practices combining software engineering, control engineering, mechanical engineering, and electrical engineering. Therefore, modeling embedded systems often results in a mix of models from a multitude of disciplines. An integrated modelling approach is provided by SysML as an extension of the Unified Modeling Language (UML), version 2, which has become the de facto standard software modeling language. SysML is a robust language that addresses many of the embedded software engineering needs, while enabling the embedded software engineering community to leverage the broad base of experience and tool vendors that support UML. Embedded systems are often safety-critical applications where correct operation is vital to ensure the safety of the public and environment. Furthermore, these systems have to fulfill real-time requirements and they have to cope with restricted resources Finally, we focus on several development processes of embedded systems and their underlying tools.

In addition to the lecture exercises are organized to give an insight how to use state of the art approaches and tools. Within small projects the students can contribute the gained knowledge by using these introduced tools and concepts.

3 Course Structure

1. Characteristics of Embedded (and real-time) Systems
2. Motivation for Embedded Software Engineering
3. Modeling of Embedded Systems
4. Overview and Architecture of SysML
    a. SysML: Requirements and Use Cases
    b. SysML: Basic Concepts
    c. SysML: Modeling Structure with Blocks
    d. SysML: Modeling Constraints with Parametrics
    e. SysML: Modeling Control Flow-Based Behavior with Activities
    f. SysML: Modeling Message-Based Behavior with Interactions
    g. SysML: Modeling Event-Based Behavior with State Machines
    h. SysML Tools in General and Enterprise Architect
5. Development Processes of Embedded Software Systems
6. SW Quality Management, Software-Test
7. Development Tools (e.g. Enterprise Architect, IBM Rational Rhapsody)

4 Parameters

- Course characteristics: compulsory
- Course frequency: every year - winter semester
- Capacity: 25 students
- Course admittance prerequisites: computer science & programming

	• Skills trained in this course: theoretical, practical and methodological skills • Assessment of the course: Written Exam (60 min) at the end of the course (50%) and group work as homework (50%) with Enterprise Architect or IBM Rhapsody use case and demonstration/presentation • Teaching staff: Prof. Dr. Stefan Henkler, (Prof. Dr. Martin Hirsch)
5	**Learning outcomes** 5.1 Knowledge • Students know the characteristics of embedded (and real-time) systems • Students know the most important SysML diagrams. • Students know the syntax and semantic of the most important SysML diagrams. • Students know modeling tools for embedded software systems. • Students know processes and methods of embedded software engineering. 5.2 Skills • Students can choose SysML-Diagrams to model specific software aspects. • Students can model structural aspects by means of block diagrams. • Students can model constraints by means of parametric diagrams. • Students can model control flow-based behavior by means of activity diagrams. • Students can model message-based behavior by means of interaction diagrams. • Students can model event-based behavior by means of state machines. • Student can tailor processes and methods to specific project needs. • Students can evaluate and use tools for embedded Software engineering. 5.3 Competence - attitude • Students develop an attitude to embedded software engineering according to modeling and processes. • Students show a quality attitude according to embedded software engineering modeling. • Students understand the main challenges of complex embedded software projects. • Students understand the importance of modeling complex embedded software systems • Students can improve their effectiveness and efficiency by using dedicated methods and tools to support engineering processes. • Students understand the differences between software and embedded software systems projects and act accordingly
6	**Teaching and training methods** • Lectures introducing concepts, methods and tools • Group work to train concepts and methods, to develop skills and to work on case studies • Home work to add contributions on a case study as group work • Presentations to communicate results

7	**Course mapping** Input for: • MOD2-01- Mechatronic Systems Engineering • MOD2-02 – Microelectronics & HW/SW Codesign • MOD-E03 – SW Architectures for Embedded and Mechatronic Systems • MOD-E10 – Automotive Systems Connects to: • MOD1-02- Distributed and Parallel Systems
8	**References** Alt, O.: Modellbasierte Systementwicklung mit SysML: in der Praxis, Carl Hanser Verlag GmbH & Co. KG, März 2012, ISBN: 978-3446430662 Friedenthal, S.; Moore, A.; Steiner, R.: A Practical Guide to SysML: The Systems Modeling Language, Morgan Kaufmann, 2nd Edition, Oktober 2011, ISBN: 978-0123852069 Oshana, R.: Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications (Expert Guide), Newnes, Mai 2013, ISBN: 978-0124159174