The SPES Methodology
Modeling- and Analysis Techniques

Dr. Wolfgang Böhm
Technische Universität München
boehmw@in.tum.de
Agenda

- SPES_XT Project Overview
- Some Basic Notions
- The SPES Methodology
- SPES_XT Engineering Challenge: Early Validation
- Demo
Agenda

• SPES_XT Project Overview

• Some Basic Notions

• The SPES Methodology

• SPES_XT Engineering Challenge: Early Validation

• Demo
SPES_XT – Software Plattform Embedded Systems

• SPES_XT is a research project funded by BMBF to develop a seamless methodology for the model based development of embedded systems.

• Successor of the innovation alliance „Software Platform Embedded Systems 2020“

• 21 partner from industry and academia

• Project budget: 25,7 Mio EUR

• www.spes2020.de
Vision of the SPES_XT Project

• Development of an integrated reference platform (methodology) based on the results of SPES2020 to support answers to essential engineering challenges in the industrial development of embedded systems.

• Development of tools to implement the methodology.

• Mapping of the methodology and techniques to selected application domains.

• Focus on practical deployment of the SPES_XT results.

• Transfer into industrial practice.
  – Approach needs to be accepted by the engineers.
  – Considering domain specific processes and techniques.
  – Practical guidance and tool support.
  – Practical use cases as prove of concepts.
SPES_XT Project Setup

Engineering Challenges (EC1 – EC6)

Cross-Cutting Topics (QT1 – QT4)

Optimal Deployment
Variant Management
Validation in early Phases

Modular Safety Case
Mechatronic and Software
Networks

Integrated and seamless methodology
Measurement of quality and efficiency
Tool platform
Transfer into practice

Application Domains (AD1 – AD3)

Automation
Automotive
Avionic
Agenda

• SPES_XT Project Overview

• Some Basic Notions

• The SPES Methodology

• SPES_XT Engineering Challenge: Early Validation

• Demo
What is Seamless Model-Based Development?

- Development by a chain of models
  - All aspects in the development captured by models
  - Clear structure of models
  - Tight integration of the models
  - Expressive power (relevant properties must be expressible)

- Development steps by well-defined relationships between models
  - Viewpoints
  - Refinement
  - Decomposition

- Extended tool support
  - High automation
  - All artifacts in tools and comprehensive data base (development back bone)

- Analyses based on models
  - Validation and verification of requirements
  - Simulation, verification
  - Virtual integration testing
System and its Context

Environment

Operational Context

Physical World

System

Internal Structure

HMI

System Boundary

Cyberspace Services & Data

Interface
- syntax
- behavior

Context System

Environment

Operational Context

Physical World

System

Internal Structure

HMI

System Boundary

Cyberspace Services & Data

Interface
- syntax
- behavior

Context System
Agenda

• SPES_XT Project Overview

• Some Basic Notions

• The SPES Methodology

• SPES_XT Engineering Challenge: Early Validation

• Demo
The following principles drive the design of the SPES Modeling Framework:

• Principle 1: **Separation of concerns**
  – Enables separating the different concerns of the stakeholders during the engineering

• Principle 2: (Hierarchical) **Decomposition**
  – To master the complexity of engineering activities

• Principle 3: **Seamless model-based engineering**
  – Establishes a continuous model-based documentation i.e. a specification of all information that is created during the different engineering activities

• Principle 4: **Separation between problem and solution**
  – Distinguishes between the analysis of the underlying problem and the construction of an appropriate solution

• Principle 4: **Separation between logical and technical solution**
  – Separates the logical solution concepts and corresponding conceptual properties from technological constraints and technological design decisions

• Principle 7: Continuously engineering of **crosscutting system properties**
  – Considers cross-cutting properties of the system-under-development
The SPES Modeling Framework

Layers of Granularity

Requirements Viewpoint  Functional Viewpoint  Logical Viewpoint  Technical Viewpoint

continuity

continuity

continuity

Viewpoints

W. Böhm
SPES Modeling Framework
Requirements Viewpoint

Models:
- Context Model
- Goal Model
- Scenario Models (SysML Sequence Diagram)
- Structural Model (Data structures - SysML Block Diagram)
- Operational Model (SysML Activities Diagram)
- Behavioral Requirements Model
- Requirements Formalization

Concerns of Requirements VP:
- Gain a **comprehensive understanding** of the system under development
- Supply necessary **information for implementation decisions**
- Foster best possible **freedom for development**
SPES Modeling Framework
Functional Viewpoint

Concerns of Functional VP:
- Consolidate functional requirements by specifying system behavior from a black box point of view
- Reduce complexity by hierarchically structuring of the functionality
- Mastering feature interaction

Models:
- Functional Black-Box Model
  - Syntactic Interface – Typed Input / Output Ports
  - Semantic Interface (Behavior: Code, State Machines)
- Function Hierarchy
- Functional White-Box Model
  - Decomposition of User Functions
  - Abstract Description of the Realization
**SPES Modeling Framework**

**Logical Viewpoint**

**Concerns of Logical VP:**
- Describe the *internal logical structure* of the SuD by partitioning the system into communicating components
- Allocating functions to logical units
- Supporting *reuse* of existing components
- Defining total system behavior

**Models:**
- Logical Component Architecture
  - Typed Ports and Communication Channels
  - Semantic Interface (Behavior)
  - Behavior of non-atomic Components derived by composition

**Analyses**
- Simulation
- Property Verification and Tests
SPES Modeling Framework
Technical Viewpoint

Concerns of Technical VP:
- Describe the resources available in the system (computation, communication, sensors, actors, …)
- Mapping resource consuming and resource offering entities (deployment)
- Scheduling

Models:
- Electric / Electronic Architecture
  - Abstraction from platform resources
  - Network of computing resources connected by communication resources
  - Resources present the allocated behavior of the logical architecture
  - Data encapsulation
Agenda

• SPES_XT Project Overview

• Some Basic Notions

• The SPES Methodology

• SPES_XT Engineering Challenge: Early Validation

• Demo
Early Validation - Goals

• Improve customer satisfaction by early fault detection in the development of embedded systems
  – Increase objectivity
  – Freedom from contradiction
  – Completeness

• Simulation framework for early validation of development artifacts
  – Modeling and formalization of functional and non-functional requirements
  – Traceability across development artifacts
  – Handle complexity and scale to industrial size projects
  – Integration of software and mechatronic
Elements of the Requirements Quality Assessment Framework

- Task renders artifact
- Task contributes to validation
- Dependency between artifacts
Elements of the Requirements Quality Assessment Framework

Artifacts
- Assumptions & Context Model
- (Model-based) Requirement Artifacts

Quality Properties
- Validation Objective

Tasks
- (Project-Specific) Quality Property Definition
- Validation Artifact Selection
- Assumption & Context Modeling
- Definition of Validation Objective

Define purpose of the validation

Completeness
Consistency
Testability
Necessity
Correctness
Identifiability
Elements of the Requirements Quality Assessment Framework

Artifacts
- Assumptions & Context Model
- (Model-based) Requirement Artifacts
- Quality Properties
- Validation Objective

Tasks
- (Project-Specific) Quality Property Definition
- Validation Artifact Selection
- Definition of Validation Objective
- Assumption & Context Modeling
- Requirements Validation (Technique)

Validation Artifacts
- Natural Language Requirements
- Semi-Formal Requirements
- Formal Requirements
- Domain-Ontology
- Product-Ontology

Supporting Artifacts
- Fagan Inspection
- Validation against stakeholder knowledge
- Multi-Aspect Analyses (Safety + Timing)
- (Co-)Simulation

W. Böhm
Elements of the Requirements Quality Assessment Framework

W. Böhm
Demo Example: Desalination Plant

Goals:
• Development of executable requirements models and formal context description for simulative validation

Approach:
• Using SPES methodology to model the desalination plant with focus on automation software
• Requirements modeling starting with use cases via scenarios to executable models
• Context modeling
  – Technical Process
  – Operator
  – Validation scenarios
• Validation: Co-simulation of process and automation
Early Validation of Automation Software Using Simulation

Mathematical models of physical plant component behavior

Simulated Technical Plant Processes

Simulation

Executed Automation Software Behavior

Revealed Defects in Requirements of the Control Software

Executable Functional Specification

Prospective Technical Plant Architecture

Assumptions in Context Models

Validation Use Cases for Simulation

process requirements

Document explicitly

Develop systematically

Simulation output

input for simulation

simulation output

input for simulation

Stepwise refinement & transformation

Key

Engineering Progress

Simulation Reference

Activity

Artifact
Early Validation: Kontext Modeling

- Structural context
- Functional context
- Context behavior

Interface definition
Creation of validation scenarios
Definition of behavior specification

Mathematical models of physical plant component behavior

Structural Operational Context Model
Functional Operational Context Model
Behavioral Operational Context Model
Creation of validation use cases from formal requirements

<table>
<thead>
<tr>
<th>Title</th>
<th>Start Beach Well</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>A beach well is manually started by the user of the desalination plant</td>
</tr>
<tr>
<td>Trigger</td>
<td>BWC Control == &quot;The user initiates the start of the beach well&quot;</td>
</tr>
<tr>
<td>Precondition</td>
<td>bypass_valve.open == true &amp;&amp; discharge_valve.closed == true</td>
</tr>
<tr>
<td>Postcondition</td>
<td>pump.on == true &amp;&amp; discharge_valve.open == true &amp;&amp; bypass_valve.closed == true</td>
</tr>
<tr>
<td>Step</td>
<td>Action</td>
</tr>
<tr>
<td>1</td>
<td>The User initiates the start of a beach well</td>
</tr>
<tr>
<td>2</td>
<td>The Beach Well Software checks whether the beach well is in standby</td>
</tr>
<tr>
<td>3</td>
<td>The Beach Well Software closes the discharge valve</td>
</tr>
<tr>
<td>4</td>
<td>The Discharge Valve sends feedback that the valve is closed</td>
</tr>
<tr>
<td>5</td>
<td>The Beach Well Software starts the Pump with minimal revolutions</td>
</tr>
<tr>
<td>6</td>
<td>The Pump sends feedback that the Pump is started</td>
</tr>
<tr>
<td>7</td>
<td>The Beach Well Software closes the Bypass valve</td>
</tr>
<tr>
<td>8</td>
<td>The Beach Well Software opens the Discharge Valve</td>
</tr>
<tr>
<td>9</td>
<td>The Discharge Valve sends feedback that the valve is opened</td>
</tr>
<tr>
<td>10</td>
<td>The Beach Well Software reports that the beach well is on</td>
</tr>
</tbody>
</table>
Linking formalized requirements (state-machines) to a simulation model of the technical process (algebraic equations)
Agenda

• SPES_XT Project Overview

• Some Basic Notions

• The SPES Methodology

• SPES_XT Engineering Challenge: Early Validation

• Demo

  Dr. Andreas Vogelsang (TUM)
  Dr. Jan-Christoph Wehrstedt (Siemens)
Questions ?