System Modelling

Foundations

- What is a model?
  - Abstraction, neglecting irrelevant parts/aspects
  - Existing systems, planned systems
  - Substitution for the modelled system
  - Reality vs. model vs. diagram
- „Closed world" vs. „open world" assumption
- Models and modelling languages
  - Syntax (abstract and concrete)
  - Semantics
  - Conditions, constraints (well-formedness)
  - Modelling, metamodelling
- Goal/Purpose of modelling
- System, context, interface
  - Black box modelling, white box modelling
  - Components, interactions
- Model refinement, model abstraction
  - Refinement of the "boxes", refinement of value sets, …
  - Refinement: adding more details
  - Abstraction: Generalization, reducing model "size"

Structural Modelling

- Structural models
  - "part of" relations, decomposition
  - Subsystems, components
  - Parts, properties, relationships
- Common models, views of a model
- Structural decomposition
  - Multi-level decomposition
  - Physical decomposition, logical decomposition
  - Correctness of a decomposition
- "Top-down" and "bottom-up" approaches
- Graph representation of structural models
  - Containment relation → tree graphs
  - Graph operations (filtering on edge labels, …)
- Property modelling
  - Property as a function
  - Tabular representation

    - Filtering and projection
        - Queries on data structures
- Modelling of types
    - Types and attributes
    - Type hierarchies (inheritance, specialization, abstraction)
    - Metamodel, instance model
        - Type nodes, edge nodes
        - Consistent modelling

State Based Modelling

- Behavioural models
    - What are the properties of the system now, and how do they „change"?
    - Behaviour fully determined by the "past" only?
- Events
    - Event streams, event spaces, series of (instantaneous) events
- State spaces
    - Discrete states, finite state spaces, current state, state refinement/abstraction
    - Completeness, mutual exclusivity
    - Composition of state spaces
        - Direct product, state vectors
        - Projection to the components (abstraction)
        - Refinement of the composition: not all combinations can manifest
            - Refinement resulting in less elements
        - Decomposition of state variables
- State transitions
    - Instantaneous transitions, discrete events
    - Binary relation
    - State graphs (Complete graphs? Reachability of states?)
    - Non-determinism, (potential) conflict of transitions
    - Labelling transitions with events
        - Pre-conditions, post-conditions, guards
        - Spontaneous transitions
        - Reading from multiple input channels, writing to multiple output channels
        - Unspecified inputs (3 approaches)
- State machine extensions
    - State hierarchy
        - Sub-states, super states (emphasizing common properties)
        - State configurations
    - Orthogonality
        - Regions, "parallel execution"
        - State configurations
        - Guards referring to states of the other regions
    - Variables
        - E.g. counters (as natural numbers? as integers?)

        - Separate regions (With their own states. Can be referred to in guards.)
    - Pseudo states
        - Syntactically states, but semantically not
- Products
    - (Direct product of state spaces)
    - Asynchronous product of state machines
    - Synchronous product of state machines
    - Mixed product of state machines
        - Rendezvous events

Process Modelling

- Behavioural models
    - What “does” the system with its working item?
    - As a series of activities
        - control flow
        - data flow
- The role of process modelling
    - In specification/design/implementation/verification/documentation
- Modelling aspects
    - Goal/output of the process
    - Activities, their order/dependencies
    - Decision points
    - Pools, roles, resources
- Uses of process models
    - Modelling administrative (e.g. banking) processes
    - Modelling manufacturing processes
    - Modelling business processes
    - Modelling operation of (e.g. IT) systems
    - Specifying protocols
    - Designing executable processes
    - Designing data processing/analysing processes
- Basic concepts
    - Process description languages
    - Process models (templates)
        - Control flow, data flow, decision points
        - Data structures
        - Steps to execute (as activities)
        - Timings, resources, roles
    - Process instances
- Basic concepts of process modelling languages
    - Elementary activity
    - Sequence
    - Decision-Merge
        - Guards, branches, control elements

    - Loop
    - Parallel execution / Fork-Join
        - Unspecified order
    - Flow begin, flow end
    - Hierarchy
    - References / Calls
- Well-structured processes
- Separation of control logic and resources
- Control flow (of programs)
    - Cyclomatic complexity of code
    - Recursion
- Process Execution
    - States of an elementary activity
    - States of a process
    - Mathematical background: Allen's interval algebra
    - Workflow mining
- Business process modelling
    - BPMN, UML AD, …
    - States of an activity
    - Events, timers, …
- Execution of process models
    - Workflow engines

Model V&V

- Model life cycle
    - Comparison to SW development life cycle
    - Life cycle of model-based SW development
- Model and activities
    - Synthesis, Analysis, Control
- Correctness (of models/implementations)
    - Fulfilment of the requirements
        - Functional requirements
            - Allowed behaviour, expected behaviour
        - Non-functional requirements
    - Deadlock, livelock
- Types of the analysis
    - Verification, Validation
    - Static analysis
        - Analysis of the structure of the model
            - Well-structured process models, well-structured code
            - Structural correctness
        - Analysis of the data flow
            - Symbolic execution
        - Syntax analysis
        - Support of design rules/patterns

- Dynamic analysis
  - Spot checking: testing, simulation
    - Test executor, test oracle, reference
    - Self testing (monitoring)
      - Exceptions, assertions
    - Coverages of test suites
      - State coverage, transition coverage
      - Statement coverage, branch coverage, path coverage, …
    - Usage of the tested models
      - SW testing, monitoring, log analysis
    - Test documentation
      - Test specification, test report
    - Phases of testing
      - Module/unit testing, integration test, system test, regression test
  - Complete checking: Formal verification
    - Proving correctness by mathematical methods
    - Model checking
      - Exhaustive analysis of possible behaviour
      - Searching for counter examples
    - Automatic proof of correctness
    - Conformance testing