Modular Functional Specifications
or: How to break down complex things to build them up orderly

Bernhard Schätz
Technische Universität München, Institut für Informatik

Univerzita Karlova Praha
Prague, February 13th, 2008
Chair Software & Systems Engineering

- Competence Centers

- Software Architecture & Components
- Model Driven Software Engineering
- IT Security

- Embedded Systems
- System Development Processes
- Theorem Proving

- Mobility & Context Aware Architecture
- Requirements Engineering
- Tool Support

- Software Maintenance
Overview

Motivation
Functions: Modular Behavior
Composition: Structured Behavior
Application: Complex Behavior
Conclusion
Motivation

• Models in the development process

Functional Architecture

Logical Architecture

Technical Architecture
Structured Development
Structured Development of Functional Specifications

Structured, descriptive, modular
Overview

Motivation

Functions: Modular Behavior
Composition: Structured Behavior
Implementation: Refined Behavior
Conclusion
Data Flow Interface

Data Flow Interactions of Functions:

- Point of Interaction: Port (input/output variable)
- Kind of Interaction: Sending and receiving of values
Observation: Describing Data Flow of Functions

- **State:** Assignment of values to ports
- **Observation:** Sequence of states
Control Flow Interface

Control Flow Interactions:
• Point of Interaction: Locations (entry/exit)
• Kind of Interaction: Activation and deactivation
Behavior: Describing Data Flow/Control Flow of Function

- Execution: Observation between activation and deactivation
- Behavior: Collection of all possible executions
Overview

Motivation
Functions: Modular Behavior
**Composition: Structured Behavior**
Application: Complex Behavior
Conclusion
Basis: Transitions

Transitions as Basic Functions:
- Input assignments: Interpreted over pre-state
- Output assignments: Interpreted over post-state
Basis: Transitions

Behavior of Transitions:
• Execution: Respecting pre-input, post-output
• Behavior: Collection of all executions
Composition: Disjunction

Disjunctive composition of functions
• Interface: Combined interfaces of functions
• Behavior: Combined behavior of functions
Composition: Conjunction

Conjunctive Composition of Functions
• Interface: Shared Interface of Functions
• Behavior: Shared Behavior of Functions
Composition: Structuring Behavior

• Disjunctive Composition:
  – Alternative execution of functions
  – Extension of behavior: Introduction of under-specification by non-determinism

• Conjunctive Composition:
  – Simultaneous execution of functions
  – Restriction of behavior: Introduction of over-specification by partiality
Composition: Hiding and Renaming

Further composition operations:
• Hide locations: Compose functions by internal deactivation/activation
• Hide ports: Compose functions by internal send/receive
• Rename locations: Adapt control interface
• Rename port: Adapt data interface
Overview

Motivation
Functions: Modular Behavior
Composition: Structured Behavior
**Application: Complex Behavior**
Conclusion
Combination: Structuring Behavior

Structured description:

- Uses modules of basic functionalities
- Combines functionalities to form more complex functionalities
- Applies conjunctive and disjunctive combination of functions
Basic window: Individual functionality

- Window movement initiated by button (Up, Dn)
- Window movement continued after release (Hd)
- Window movement aborted by button (Dn, Up)
- Symmetrical for upward/downward movement
Modules: Position Control

Position Control: Alternative individual functionalities

- Position Check: Movement restricted to intermediate positions
- Position Override: Movement aborted by end position detection
- Symmetrical for upward/downward movement
Power Control: Alternative individual functionalities
- Power Check: Activation requires sufficient voltage
- Power Override: Lack of voltage blocks activation
Combination: Position over Basic Window

Combination of modules: „Check-and-Override“-Pattern

- Conjunctive Combination: Simultaneous behavior
  - Position Check: Movement unrestricted in intermediate position
  - Basic Window: Basic window movement

- Disjunctive Combination: Alternative behavior
  - Control Window: Behavior in case of intermediate position
  - Position Override: Activation stopped by end position detection
Combination of modules: „Delegate“-Pattern

- Disjunctive Combination: Alternative behavior
  - Position Down: Position controlled downward window movement, including hold of movement
  - Position Up: Position controlled upward window movement, including hold of movement
Combination: Power over Position

Combination of modules: „Check-and-override“-Pattern
- Disjunctive Combination: Alternative behavior
  - Power Override: Activation blocked by lack of voltage
  - Controlled Position: Behavior in case of sufficient voltage
- Conjunctive Combination: Simultaneous behavior
  - Power Check: Movement supported by sufficient voltage
  - Position Window: Position controlled window movement
Overview

Motivation
Functions: Modular Behavior
Composition: Structured Behavior
Implementation: Refined Behavior
Conclusion
Implementation: Refinement by Restriction

Implementation:
• Refinement: Restriction of under-specified behavior
• Restriction: Elimination of non-determinism
Implementation: Refinement by Extension

Implementation:
- Refinement: Extension of un-specified behavior
- Extension: Elimination of partiality
Implementability:
- Function: Potentially underspecified/partial behavior
- Partiality: Undefined system reactions for some environment actions
- Implementable behavior: Defined system reactions for all environment actions
Implementation: Implementability and Refinement

Implementation:
• Check Implementability: No partiality left
• Check Refinement: All behavior included
• Automatic Verification: Model Checking (MONA)
Overview

Motivation
Functions: Modular Behavior
Composition: Structured Behavior
Implementation: Refined Behavior

Conclusion
Conclusion

Functional modules: Modules of control flow and data flow

- Support modular description of functionality
- Allow reduction of complexity functionality
- Support structured description of functionality
- Allow compositional reasoning about functionality
- Support compositional description of functionality
- Allow refinement to constructive description