Early Prototyping of Reactive Systems
Through the Generation of SDL Specifications
from Semi-formal Development Documents

Andreas Metzger, Stefan Queins
Department of Computer Science
University of Kaiserslautern, Germany

Abstract Concepts

Concrete Concepts

Example

Motivation

Conclusion and Perspectives
Motivation

Application of Prototyping

Idea

Validation (Product Test) → Problem Analysis

Problem Description

Verification (e.g. Inspection) → Requirements Analysis

Requirements Specification

Final System

Validation (Prototype Test)

Motivation

Abstract Concepts

Requirements Engineering

Concrete Concepts

Development Products

Example

Conclusion and Perspectives
Efficient Creation of Prototypes

1. Idea
2. Problem Analysis
3. Problem Description
4. Requirements Analysis
5. Requirements Specification
6. Prototype Generation
7. System Prototype
8. Final System

Early Generation of Prototypes

1. Problem Description
2. Requirements Description
3. Semi-formal Requirements Specification
4. Requirements Modeling
5. Operational (Formal) Requirements Specification
6. Operational Specification Generation
7. Product Model
8. Final System
9. System Prototype
Development Products

Classification of Product Model Entities

Early Generation of Prototypes

Application of Product Model Entities
**Early Generation of Prototypes**

**Operational Specification Generation**

![Diagram showing the process of early generation of prototypes]

**Benefits and Drawbacks Compared to Generic Generators**

**+ less formal specification** sufficient

  - **less** specification **effort** required

  - **earlier** prototype **generation** possible

**+ “effortless” integration of new notations**

  - creation of new feature view parsers or generators

**– precise product model** needed

  - generic specification method insufficient

  - creation of specific generators/parsers for each kind of product model required
Early Prototyping of Reactive Systems...

The PROBAнд Method

"Prototyping-, Reuse- & Object-based Building Automation Development"

Motivation

Abstract Concepts

Requirements Engineering

Concrete Concepts

Development Products

The PROBAнд Method

The Product Model of PROBAнд

Example

Conclusion and Perspectives
The PROBAnD Method

SDL Document Generation

Building Description

Problem Description

Needs

C. S. Structure Specification

Control System Description

Control Task Description

Control System Structure (HTML)

Semi-formal Control Object Type (HTML)

Control Tasks (HTML)

Control System Modeling

Operational Control Object Type (SDL)

Formal (Operational) Requirements Specification

Prototype Generation

Control System Prototype

SDL Specification Generation

Product Model

The Product Model of PROBAnD

Types of Features (Excerpt)

Requirement

realizedBy

Need

realizedBy

Control Task

implementedBy

Control Object Type

Strategy

Instantiation

aggregates

controls

Attribute

writes

reads

produces

consumes

Parameter

Signal Type

Functional Strategy


9

10
Early Generation of Prototypes

Application of the Product Model of PROBAnD

SDL Control Object Type Document

SDL Feature View Generation

SDL Composition

SDL Control Object Type
SDL Strategy
SDL Attribute
SDL Instantiation
SDL Parameter
SDL Signal Type

Features and Configurations

Semi-Formal Document (HTML)

Operational Control Object Type Document (SDL)

Product Model

Prototype Generation

Control System Prototype

HTML Document Parsing

SDL Document Generation

Early Prototyping of Reactive Systems...

Example: “Simple Lighting Control System”

Building Description

- 1 floor with 3 rooms
- 2 luminaires, 2 push-buttons, and 1 motion-detector per room

Control System Structure
**Example: “Simple Lighting Control System”**

### Building Description

- 1 floor with 3 rooms
- 2 luminaires, 2 push-buttons, and 1 motion-detector per room

### Control System Structure

![Diagram of control system structure]

### Needs

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>need_1</td>
<td>It should be possible to turn on the lights in a room</td>
</tr>
</tbody>
</table>

### Control Tasks

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>realized by</th>
<th>realizes</th>
<th>Control ObjectType</th>
</tr>
</thead>
<tbody>
<tr>
<td>task_1</td>
<td>Turn on luminaire if push-button is pressed</td>
<td>task_2, task_3, need_1</td>
<td></td>
<td>LightingZone</td>
</tr>
<tr>
<td>task_2</td>
<td>Notify of the push-button being pressed</td>
<td>task_1</td>
<td></td>
<td>PushButton</td>
</tr>
<tr>
<td>task_3</td>
<td>Turn luminaire on or off</td>
<td>task_1</td>
<td></td>
<td>Luminaire</td>
</tr>
</tbody>
</table>
Example: “Simple Lighting Control System”

Control Object Type: LightingZone

### HTMLControlObject

**Model Name**

- **Description**
  - Turn on luminaire if push-button is pressed

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Description</th>
<th>StrategyDescription</th>
<th>Realized Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>task_1</td>
<td>Turn on luminaire if push-button is pressed</td>
<td>If the signal <code>newPushButton</code> is received, send the signal <code>setLuminaire(on)</code> to turn on the luminaire</td>
<td>need_1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Parameters</th>
<th>Tasks</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>newPushButton</td>
<td></td>
<td>task_1</td>
<td>consumed</td>
</tr>
<tr>
<td>setLuminaire</td>
<td>boolean</td>
<td>task_1</td>
<td>produced</td>
</tr>
</tbody>
</table>

**Features**

- **strategy_1**: Functional Strategy
  - consumes
  - produces

- **newPushButton**: Signal Type
- **setLuminaire**: Signal Type

- **Luminaire**: Control Object Type
- **LightingZone**: Control Object Type
- **task_1**: Task
  - implements

HTML Document Parsing

- **HTML Object Structure Document**
- **HTML Control Object Type Document**
- **HTML Task List Document**

- **HTML Document Parsing**
  - `implemented` by task_1
  - `produced` by setLuminaire
  - `consumed` by newPushButton
Example: “Simple Lighting Control System”

SDL Document Generation

```
Example: “Simple Lighting Control System”

SDL Document Composition: Structure

Block Type LightingZone

/* Signals to subobjects via a */
SIGNALLIST ai1 = (empty);
SIGNALLIST ai2 = setLuminaire;
/* Signals from subobjects via b */
SIGNALLIST bi1 = newPushButton;
SIGNALLIST bi2 = (empty);
```

Example: “Simple Lighting Control System”

SDL Document Composition: Behavior

Realized:

+ definition and initialization of attributes
+ initialization of processes
+ signal routing derived from routing/propagation strategies

Missing:

– behavior generation from functional strategies
  • possible solutions:  - UCMs [Amyot]
    - MSCs [Mansurov]
    - Petri Nets
    - “automatic” composition of (E)FSMs

Conclusion and Perspectives

Conclusion

• systematic approach for early prototyping
  – generation of formal specifications (SDL) from semi-formal documents (HTML)
  – application of precise product model
  – use of modeling templates (reuse)

Perspectives

• iterative development

• case studies
  – quantification of gain of efficiency

• extension to other domains

• (abstract) behavior description
  – generation of behavior from “semi-formal” documents