A Software Development Process Supporting Non-functional Properties

Simone Röttger, Steffen Zschaler
COMQUAD project
Dresden University of Technology
Outline

- Motivation
- Roles in Design
- Tool-Supported Refinement
- Conclusion & Outlook
What do we want?

- **COMQUAD = Components with Quantitative Properties and Adaptivity**

- **Project objectives:**
  - development methodology and
  - runtime environment

- **This presentation: concentration on methodology and related aspects**

**Main thesis:**

Non-functional properties constitute a project risk and need to be considered as early as possible during development.
CQML+: A Language for QoS Specification

- Based on CQML (Aagedal)
quality_characteristic response_time (op: Operation) {
    domain: numeric real \([0..)\) milliseconds;
    values: let emission : Event = op.SE->last() in
        let reception: Event = op.SR->any (e | e.equals (emission)) in
            (emission.time() - reception.time()).abs()
}

quality good_response (op: Operation) {
    response_time (op) < 500;
}

profile goodCatalogResponse for VideoBooking {
    provides good_response (booking.getCatalog);
}

An example specification

Interface
<<Interface>>

Component
0..n
+usedInterface
0..n

Announcement

OperationalInterface

Interrogation

Call

Operation

1..n
+caller
1
+callee

OperationalInterface

StreamEndpoint

1..n
+flows
1..n
+endpoints
2..n

Stream

Flow

FlowEndpoint

2..n
+endpoints
2..n
+flows 1..n

<<component>> VideoBooking

<<Operation>>
getCatalog

<<Interface>>
booking
How can this be improved?

- **Specification mixes two issues:**
  - Definition of measurements

```plaintext
quality_characteristic response_time (op: Operation) {
    domain: numeric real [0..) milliseconds;
    values: ...
}
```
  - Application of measurements to constrain application model

```plaintext
quality good_response (op: Operation) {
    response_time (op) < 500;
}
```

```plaintext
profile goodCatalogResponse for VideoBooking {
    provides good_response (booking.getCatalog);
}
```
Two New Roles

Measurement Designer

Application Designer

Library

Define Measurements

Non-functional Application Modelling

Specify Constraints, Bind to Components

Environment Modelling

Specify Constraints, Bind to Connectors

Refinement

Refinement

Monitoring Code

Runtime Specification
Context Models

- Measurement Designer must be independent of concrete applications

  - Context Models capture application parts essential for definition of a measurement
    - Both structural and behavioural
    - Apply measurement to application model iff application model implementation of context model

- Context models can be defined at various levels of abstraction
  - Allows refinement support for application designer
Tool-Supported Refinement

More Abstract
- Measurement Definition
  → Context Model
  Transformation Specification

Less Abstract
- Measurement Definition
  → Context Model

Application Designer
- Non-functional Application Model
  Tool-Supported Refinement
  Context Model – Application Model Mapping

Definition of Measurements
Usage of Measurements
Conclusions & Outlook

- Separated definition of measurements from their usage
  - Measurement Designer vs. Application Designer
  - Context Models

- Provided tool-based refinement support for application designer

Future work:
- Define specification language for context model transformations
- Complete development process
- Rework and formalize specification language
The End