TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Seminar aus Programmiersprachen

Markus Raab <markus.raab@complang.tuwien.ac.at>

LVA 185.307

09.03.2016

Outline

- Problem description
- Topics

- Seminar course
- Grading

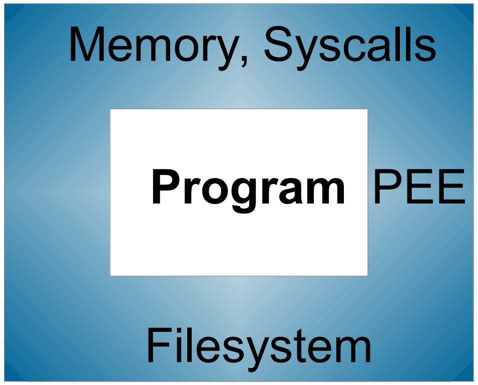
- Select date
- Choose topic+Registration

Elektra's Logo
http://www.libelektra.org

Program Execution Environment (PEE)

- Interface of every application to operating system
- Configuration management

- Consists of:
  - Configuration Files
  - Command-line Arguments
  - …



Motivation

- Program Execution Environment (PEE)
  - PEE **error-prone** on multiple levels
    - Inconsistency in documentation and semantics (behavior)
    - Hardly any validation, difficult to specify validation
    - Redundancy and wrong conversions within programs
  - Applications have **undefined behavior** on errors
  - Currently no standard way to specify PEE

Question

- How can we reduce software *failures* caused by *invalid configuration files*?

- **Failure**: deviation from desired behavior
- **Error**: incorrect state that may lead to failure
- **Fault**: cause of an error
  - Reasons of *faults*:
    - Wrong conversations
    - Improper use of values
    - Inconsistent/unexpected checks and fall backs

(Terminology: IEEE 610.12-1990)

Models of Errors

- Fuzzing/Random testing
- Use model of administrator's action
  - Typos (case, insertion, substitution, transposition)
  - Structural errors (missing sections, parameters in wrong sections, omission of necessary parameters in section)
  - Semantic errors (wrong version, documentation, confuse similar applications, domain specific misunderstanding)
- Constraint-based
  - Analyze source code and find out how parameters interact
- Semantic
  - File path, IP address, port number may be syntactically correct, but not available

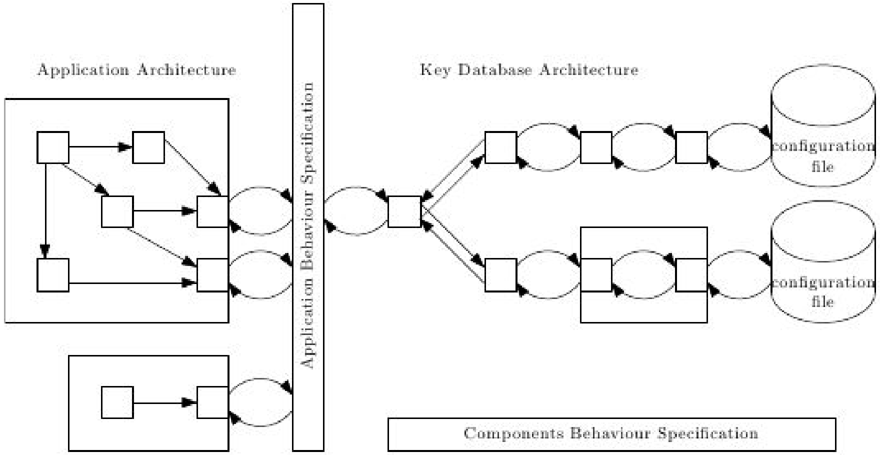
Solution

Specification Languages

- Specification
  - Validation of configuration files
  - Migrate between versions of configuration files
- Testing
  - Generate Test data from specification
- Runtime
  - Ensure that configuration file corresponds to specification
- We will look into Configuration Specification Languages
  - Different communities have different name for it

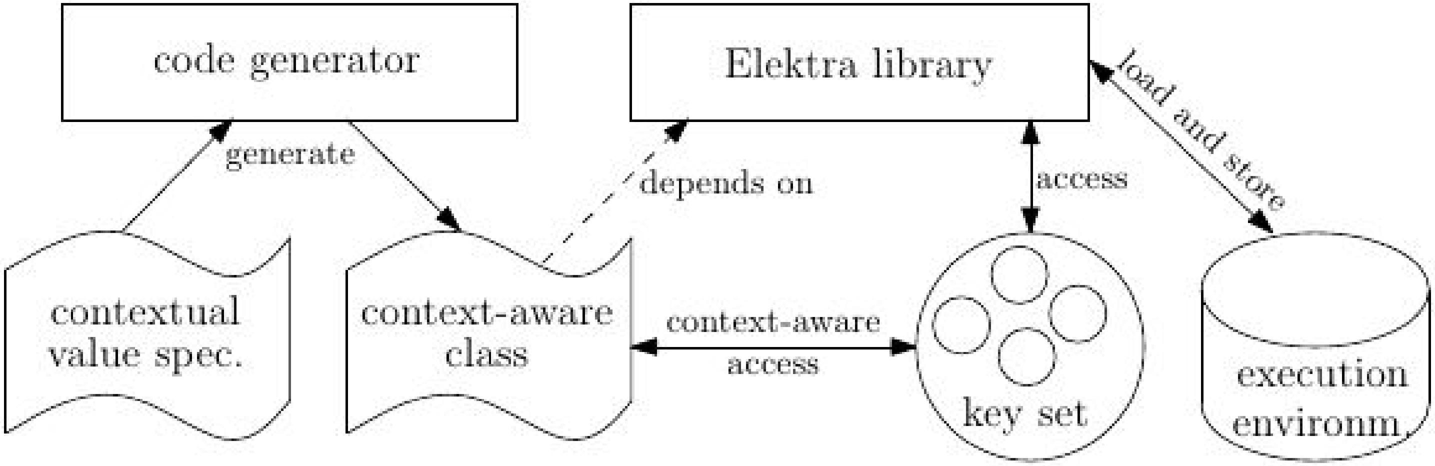
Approach

- Use a single, extensible specification
- Implement validators and checkers as plugins


Application Architecture
Application Behaviour Specification
Key Database Architecture
configuration file
configuration file
Components Behaviour Specification

Approach cont.

- Generate code and validate using specification



Seminar Course

Ablauf

- Vorbesprechung
- Abgabe Literatur Recherche
- Abgabe Paper
- Abgabe Review
- Workshop (Präsentationen)

Paper

- Umfang
  - Max. 6 Seiten (ohne Referenzen)
  - Min. 6 Referenzen
  - Kurz, aber viel Inhalt
- Intensive Literaturrecherche verlangt
  - Andere Papers
  - Evtl. Bücher
- Sonstiges
  - Englisch
  - Veröffentlichung

Praktika

- Auch Abschlussarbeiten und Praktika über diese Themen möglich
- 6 ECTS/12 ECTS Praktika
- Bitte mich anschreiben

ECTS Breakdown

- 3 ECTS=75h
  - Anwesenheit: 9h (3 Termine)
  - Literaturrecherche: 15h
  - Paper schreiben: 30h
  - Review: 6h
  - Review einarbeiten: 10h
  - Präsentation vorbereiten: 5h

Benotung

- Alle Teile müssen positiv sein
- Gewichtete Ergebnisse von:
  - Paper 60%
  - Review 15%
  - Präsentation 25%

Your Turn

Termine

- 24. März: Literatur Recherche
- 15. April: Deadline Paper Abgabe
- 22. April: Deadline Review
- 6. Mai: Finales Paper
- 9. Mai: Workshop
- 11. Mai: Workshop

Themen

- X Contextual (Context-Aware) configuration specification languages
- X Using constraint solvers in configuration specification languages
- X Use configuration management tools to specify configuration
- X Using types for representation of ad-hoc data
- X Bidirectional AST to AST transformation using lenses
- X Infer Configuration Requirements (constraints) from source-code
- X Definitions to the Configuration Problem
- X Configuration Languages for Build Systems (e.g. KConfig)
- (Variability and decision modeling)
- X Code generation from configuration specification languages
- X Comparison of Pluggable Type Systems
- X Using substitutability for checking ABI and the semantic interface

Thank you for your attention!

Markus Raab <markus.raab@complang.tuwien.ac.at>

Questions?

Markus Raab