Projects Presentation

Supervisor: Karine Arnout

2003
1. Porting Daikon to Eiffel

- Part of a general effort of improving existing components by adding contracts a posteriori.

- **Daikon:**
  - Developed by Michael Ernst (MIT).
  - Automatic generation of contracts.

- **Project:** Generate an Eiffel front-end for Daikon.

- **Goal:** Combine the strengths of Daikon and Eiffel to provide programmers with a more effective tool to contract their programs or libraries.
2. Developing a CIL parser

- Part of a general effort of improving existing components by adding contracts a posteriori.

- Automatic contract extraction:
  - Infer routine preconditions from exception cases.

- **Project**: Develop a CIL parser using Gobo Eiffel Lex and Yacc.

- **Goal**: Automatic extraction of routine preconditions from existing .NET libraries.
3. Component Quality Model (cont’d)

- Part of a general effort to develop a Component Quality Model and component validation standards.

- **Project:** Assess the quality of commercial components (EJB, .NET libraries, etc.) and extend the Component Quality Model if necessary.

- **Goal:** Certify the quality of existing components (Component Certification Center).
3. Component Quality Model

(Under development)

A: Social
A.1 Some reuse attested
A.2 Producer reputation
A.3 Published evaluations

B: Functional
B.1 Examples
B.2 Usage documentation
B.3 Preconditioned
B.4 Some postconditions
B.5 Full postconditions
B.6 Observable invariants

C: Performance
C.1 Platform spec
C.2 Ease of use
C.3 Response time
C.4 Memory occupation
C.5 Bandwidth
C.6 Availability
C.7 Security

D: Design
D.1 Consistent API rules
D.2 Strict design rules
D.3 Extensive test cases
D.4 Some proved properties
D.5 Proofs of preconditions, postconditions & invariants

E: Extension
E.1 Portable across platforms
E.2 Mechanisms for addition
E.3 Mechanisms for redefinition
E.4 User action pluggability

Chair of Software Engineering
4. Contract Wizard

- Part of a general effort of improving existing components by adding contracts a posteriori.


- Project: Update the wizard with the latest version of .NET and extend it.

- Goal: Provide .NET developers with a tool to add contracts a posteriori (public release). Maybe turn it into a Web service.
5. Eiffel SDL Multimedia Library

- **SDL:**
  - Multimedia
  - Portable
  - Free

- **Project:** Develop an Eiffel Multimedia Library on top of SDL.

- **Goal:** Provide Eiffel users with a multi-platform multimedia library. Use it in the forthcoming “Introduction to Programming” course.
6. Test Wizard

- Part of a research work trying to highlight the benefits of Design by Contract in testing.

- **Project:** Develop a tool generating test cases from classes with contracts.

- **Goal:** Provide Eiffel users with an automatic tool to generate test suites from their programs developed with DbC.
Contact information

- Interested?

- Send me an email: Karine.Arnout@inf.ethz.ch

- Come to my office: RZ-F7

- I’m looking forward to work with you 😊