ECONET Project

Behaviour Abstraction from Code

Filling the Gap between Component Specification and Implementation

Working sessions

first plan

Prague - september, 5-7 2007

Working Session Roadmap

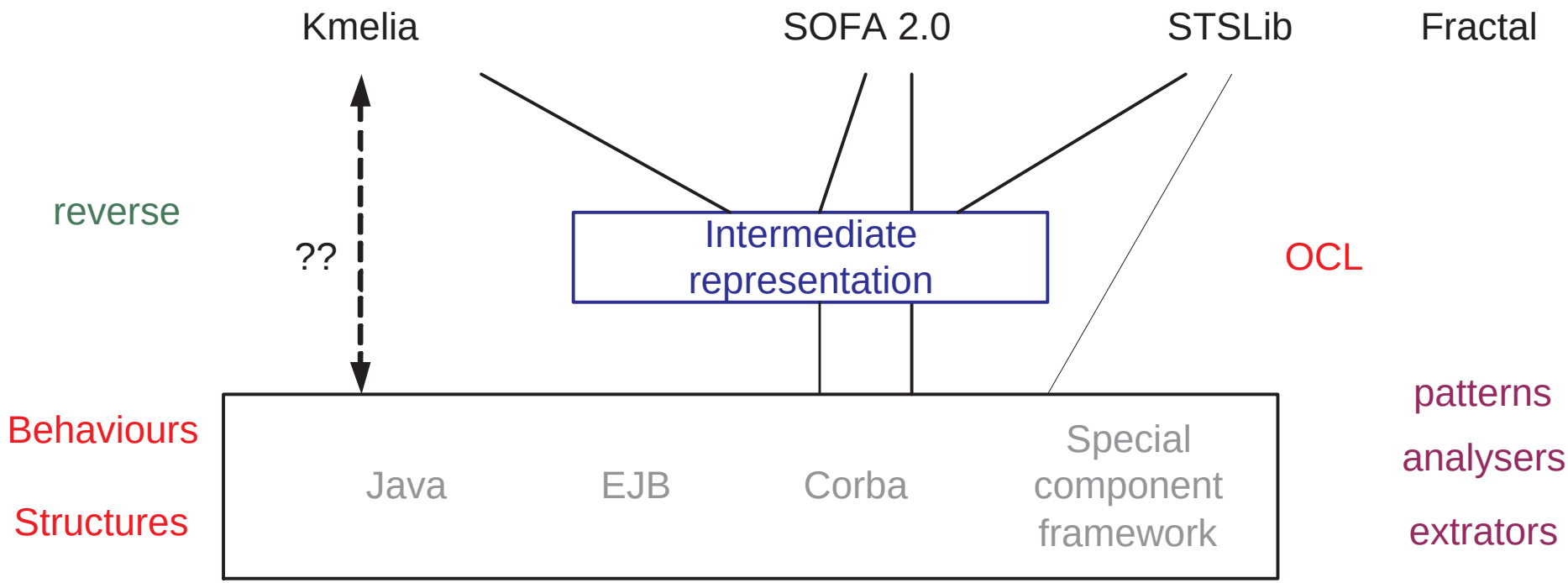
1. Convergence on the objectives
2. Convergence on the means
3. Definition of the tasks
4. Production

Convergence on the objectives

- Clear agreement on the "abstract" context

Convergence on the objectives

- Clear agreement on the "abstract" context



- Abstract component models
- Java Code
- Reverse = from code to abstract models

Convergence on the objectives

- Clear agreement on the "abstract" context
  - Abstract component models
  - Java Code
  - Reverse = from code to abstract models
- Fuzzy vision of the "concrete" context

Convergence on the objectives

- Clear agreement on the "abstract" context
  - Abstract component models
  - Java Code
  - Reverse = from code to abstract models
- Fuzzy vision of the "concrete" context
  - Java code nature
    - Bytecode
    - Plain source
    - Annoted Source

Convergence on the objectives

- Clear agreement on the "abstract" context
  - Abstract component models
  - Java Code
  - Reverse = from code to abstract models
- Fuzzy vision of the "concrete" context
  - Java code nature
  - Java code structure
    - plain Java
    - "componentised" Java (EJB, Corba, .NET, issued from a code generator - SOFA, Fractal...)
    - "behavioural" Java (threads, communication primitives, issued from a code generator...)

Convergence on the objectives

- Clear agreement on the "abstract" context
  - Abstract component models
  - Java Code
  - Reverse = from code to abstract models
- Fuzzy vision of the "concrete" context
  - Java code nature
  - Java code structure
  - reengineering issues
    - legacy code recovery/discovery
    - compare code and specifications (conformance)
    - roundtrip
    - ...

Convergence on the objectives

- Clear agreement on the "abstract" context
  - Abstract component models
  - Java Code
  - Reverse = from code to abstract models
- Fuzzy vision of the "concrete" context
  - Java code nature
  - Java code structure
  - reengineering issues

Goal of day 3 = Clear agreement on the "concrete" context

Convergence on the means

- Collaborative State of the Art Study
- Re-ingeneering techniques
  - Java Compilers and Analysers
  - Patterns, rule based systems
  - Used notations and Intermediate layers (models)
  - ...
- Separate modules (e.g. structural / behavioural / metamodels)
- Benchmark example

Convergence on the means

- Collaborative State of the Art Study
- Re-ingeneering techniques
  - Java Compilers and Analysers
  - Patterns, rule based systems
  - Used notations and Intermediate layers (models)
  - ...
- Separate modules (e.g. structural / behavioural / metamodels)
- Benchmark example

(optimistic) Goal of day 4 = organize the means tracks and find the benchmark

Definition of the tasks

- What to do ?
- Synchronisation points ?
- Planning
- ...

Definition of the tasks

- What to do ?
- Synchronisation points ?
- Planning
- ...

(optimistic) Goal of day 5 = each participant has a somewhat clear idea of what he will do

Production

- Workshop Report
  - Collect paper and slides
  - Summary of the discussions

$+$ Bibliographical Notes

$\Longrightarrow$ project plan for year 2 and *Evaluation*

- Fix the participants objectives
- Documentation, research reports
- Intermediate results $\Longrightarrow$ Second Workshop
- Publications (?)

see also the initial 'Second year objectives'

Working Session Organisation

- Plenary sessions ?
- Discussion groups ?