OVERTURE

4th Overture
Workshop

26 May 2008
Peter Gorm Larsen
Shin Sahara and
John Fitzgerald
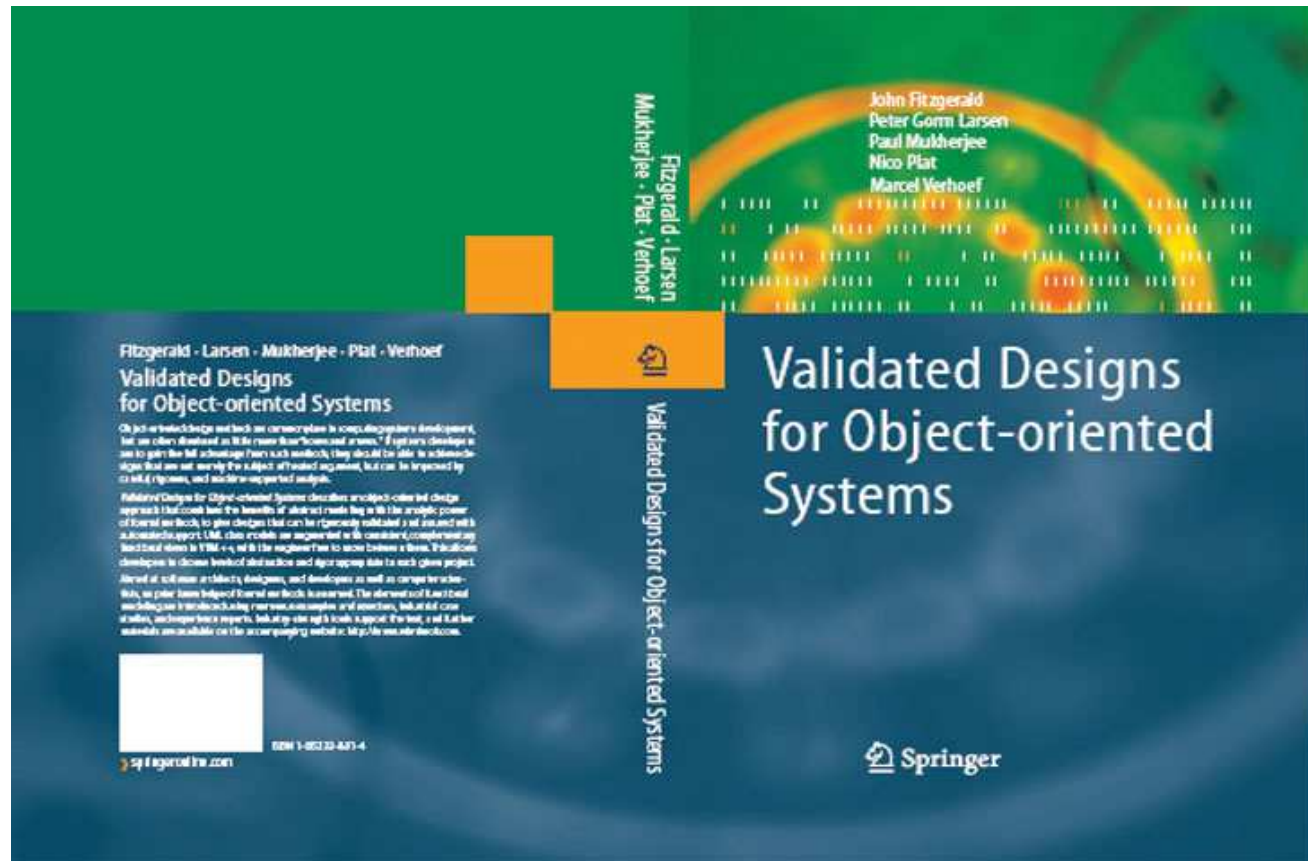
Where did it all start?

- 1992 – 1994: ESPRIT–III project AFRODITE was the start of VDM++ and an initial toolset
- 1994 – 2004: toolset and VDM++ taken up by IFAD (Denmark)
- Large customer base established: over 1000 users worldwide
- 2004: Start of Overture
- 2004: VDMTools taken over by CSK, Japan

OVERTURE

"The Book"
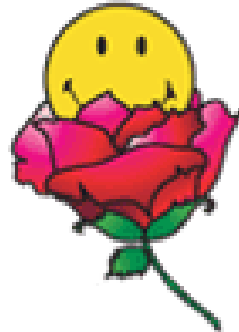
www.vdmbook.com

Then disaster hit in...

- IFAD went bankrupt (2004)
- No possibility anymore to include a "light" version of VDMTools with the book
- Since the book focuses on tool usage we had a problem.....
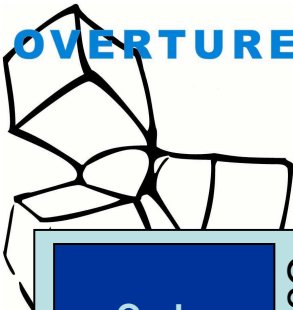
The Start of Overture

- Movement towards open source software

- New thinking, new technology in the area of IDEs and user interfaces

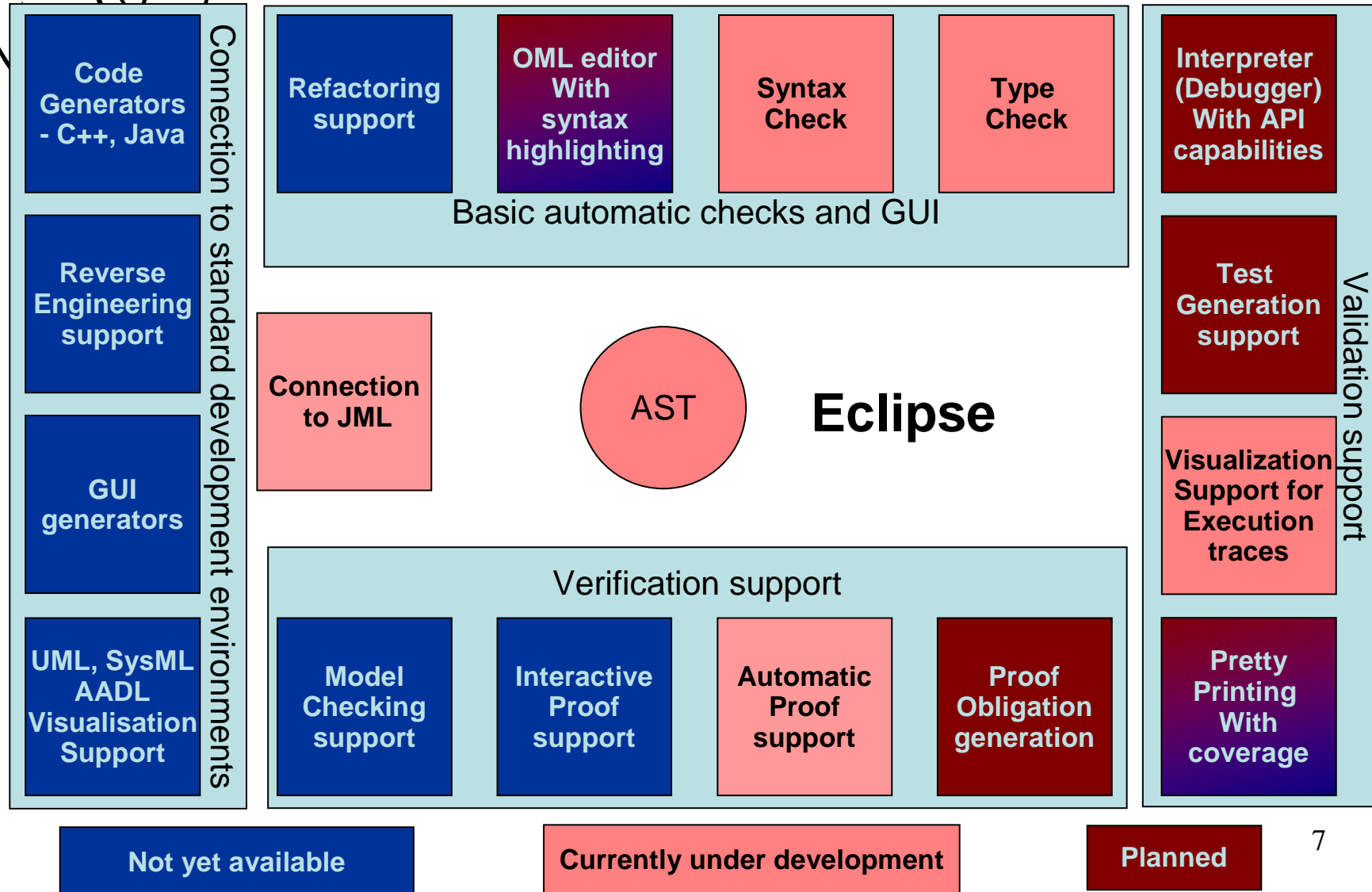- Having more tools that support VDM++ can revitalise its use

Status

OVERTURE

- Usage of open source/open standards such as Eclipse
- Basic kernel and some features available
- Monthly meetings using MSN
- Short on resources
- Student involvement
- License policy: GPL
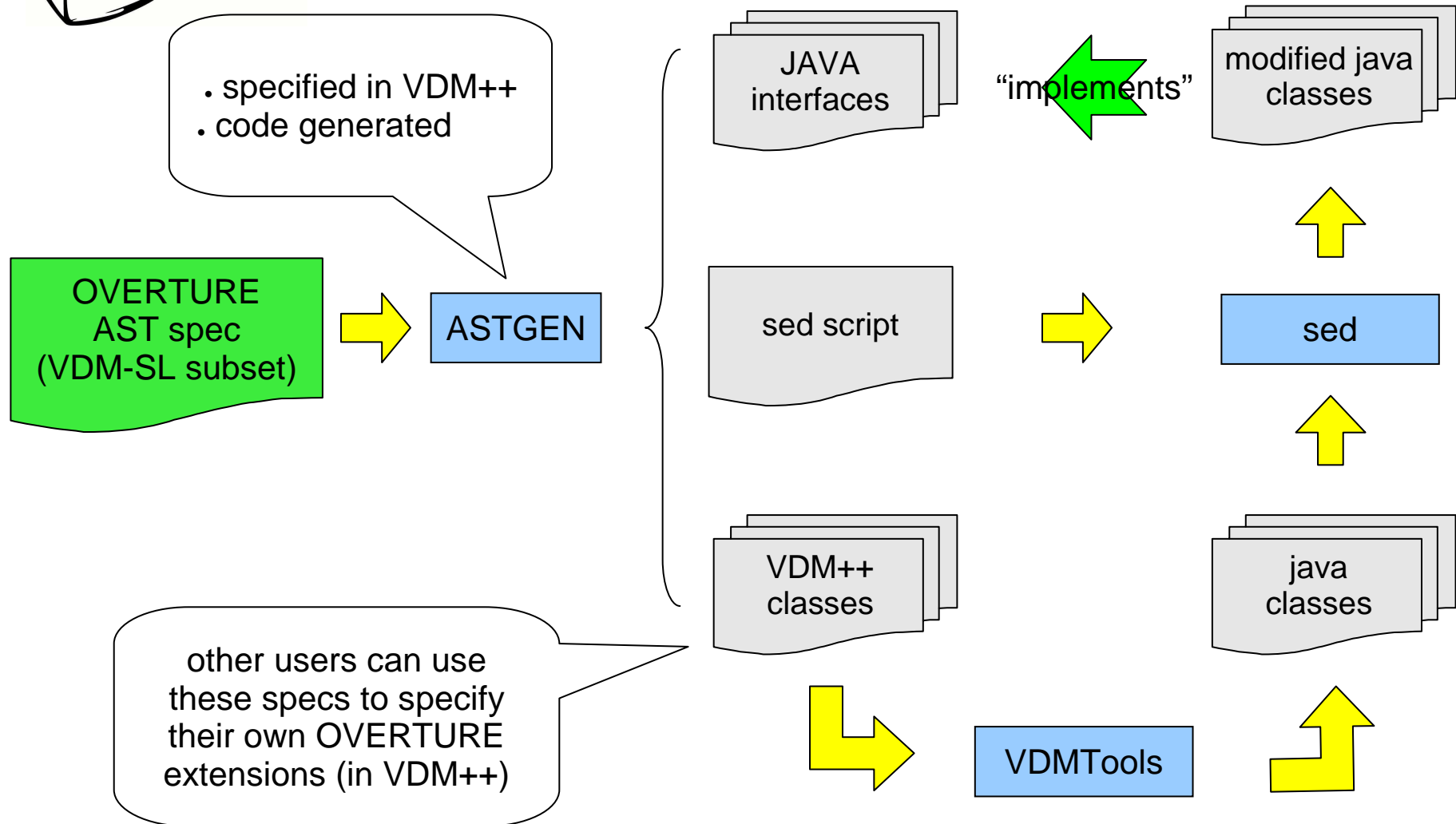- Sourceforge: http://sourceforge.net/projects/overture
- Web site: http://www.overturetool.org

Overture Architecture

OVERTURE

Code Generators - C++, Java

Reverse Engineering support

GUI generators

UML, SysML AADL Visualisation Support

Connection to standard development environments

Refactoring support

OML editor With syntax highlighting

Syntax Check

Type Check

Basic automatic checks and GUI

Connection to JML

AST

Eclipse

Verification support

Model Checking support

Interactive Proof support

Automatic Proof support

Proof Obligation generation

Interpreter (Debugger) With API capabilities

Test Generation support

Validation support

Visualization Support for Execution traces

Pretty Printing With coverage

Not yet available

Currently under development

Planned

7

Automatic AST generation

OVERTURE

- specified in VDM++
- code generated

OVERTURE AST spec (VDM-SL subset) → **ASTGEN**

JAVA interfaces

"implements" ← modified java classes

sed script → sed

VDM++ classes

other users can use these specs to specify their own OVERTURE extensions (in VDM++)

VDMTools

java classes
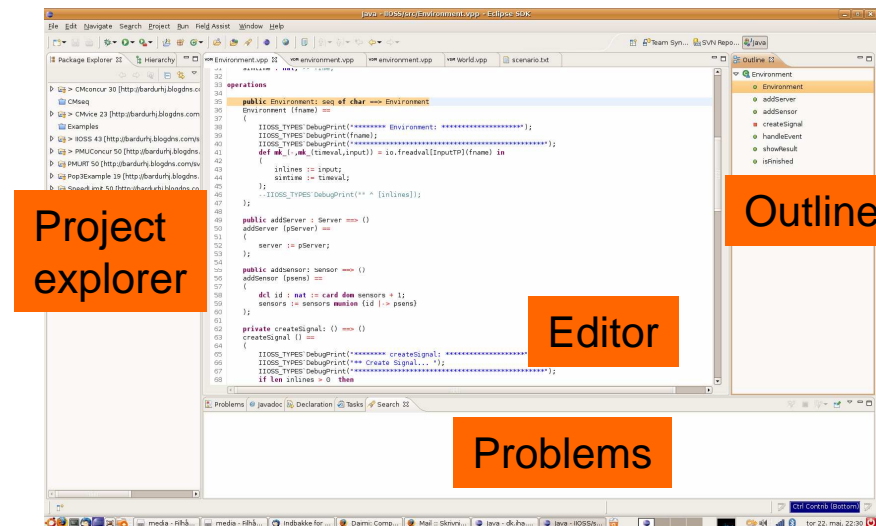
Integration Architecture

- Common eclipse interface for Overture and VDMTools
  - Modelling perspective
  - Debugging perspective (like Java)
  - Proof perspective (like Deploy)
  - Co-simulation perspective
- Move between ASTs

Agenda for the day

- Introduction
- Presentations by:
    - Volker Stolz (twice)
    - Simon Tjell
    - Carlos Vilhena
    - Adriana Santos
    - Miguel Ferreira
    - Hugo Macedo
    - Sander Vermolen
    - Marcel Verhoef
- Update on VDMTools™ by 佐原 伸
- Brainstorm session
- Joint dinner (At your own expense)

OVERTURE

Brainstorm

Potential subjects:

- H2 realise the integration architecture?

- H2 enable mapping between ASTs and maintain position information?

- When and where to arrange a new workshop with focus on tool integration

More ideas are welcome during the day!