IFAD VDMTools Alliances

- ISPRAS, Russia
- Sidereus, Portugal
- Rational, USA
- SofTools, USA
- Alagar, Canada
- Aichernig, Austria
- DDC-I, USA
- JFITS, Japan
VDMTools® Overview

- Syntax & Type Checker
- The Rose-VDM++ Link
- Document Generator
- Code Generators - C++, Java
- Interpreter (Debugger)
- API (Corba), DL Facility
Syntax and Type Checking
Validation with VDMTools®

VDM specs → Actual results

Test cases → Execution

Comparison

Expected results
Documentation in MS Word/RTF

One compound document:

- Documentation
- Specification
- Test coverage
- Test coverage statistics
The Rose-VDM++ Link

- Supports round-trip engineering with Rational Rose
- Offers the complementary benefits of the graphical notation UML and the textual formal notation VDM++
- Massive use of UML expected world-wide!
The Rose-VDM++ Link

Is my model “right”? How can I check my model?

Validate requirements and design. Test your models!

Rose-VDM++ Link

Rose2000
Integration Principle

VDM++ specification  UML Model

Mapping Rules
Associations

- Clientship relations are represented in UML as an association:

```
Producer --| buf | Buffer --| buf | Consumer
```

- class Producer
  - instance variables
    - buf: Buffer
    - ...

- class Consumer
  - instance variables
    - buf: Buffer
    - ...

- Associations can have multiplicity

```
Company |staff| * | Employee
```

- class Company
  - instance variables
    - staff: set of Employee;
    - ...

In UML inheritance is termed as \textit{generalization}.

In VDM++ the “is subclass of” keyword identifies the inheritance relations between classes.

- **Vehicle**
  - **LandVehicle**
  - **WaterVehicle**
    - **Car**
    - **Amphibious**
    - **Boat**

```plaintext
// VDM++ code

class Vehicle

  ...

end Vehicle


class LandVehicle

  is subclass of Vehicle

  ...

end LandVehicle


class Amphibious

  is subclass of LandVehicle, WaterVehicle

  ...

end Amphibious
```
A Class Diagram

The Buffer class

Attributes

- Buffer
  - <instance variable> buf : seq of Value
  - <value> size = 10

Operations

- <operation> GetItem() : Value
- <operation> PutItem(item : Value)
- <function> IncrItem(item : Value) : Value

Class Name

- <<stereotype>> name: type
- . . .

- <<stereotype>> opname(p:type,...): type
- . . .

Attributes

- <<instance variable>> buf : seq of Value
- <<value>> size = 10

Operations

- <<operation>> GetItem() : Value
- <<operation>> PutItem(item : Value)
- <<function>> IncrItem(item : Value) : Value
class A
  instance variables
toC: C
end A

class B is subclass of C
  instance variables
    b: nat;
    seqofA: seq of A
  operations
    public Get: () ==> nat
    Get() ==
      return b;
    public Set: nat ==> ()
    Set(val) ==
      b := val
end B

class C
  instance variables
    selfLink: C
end C
Architecture of Link

VDM++ Toolbox

Rational Rose 2000

Class Repository

UML Diagrams

Merge Tool

Class Repository

UML model file

VDM++ Files
Toolbox API

Request

Result

The balance of Card 4 is 5000
Dynamic Link Facility

VDM Specification

Dynamic Link Module

External Code

Type Conversion Module
Japanese Support
Free Academic Site Licenses

- For teaching purposes
- For research purposes
- So far more than 30 around the world
- Fitzgerald&Larsen book translated to Japanese
- A VDM++ book to be published 2002
Future VDMTools Extensions

- Reverse engineering from Java
- Real-time features
- Proof support
- Test case generation (ISPRAS)
- Database reverse engineering (Sidereus)
- No more EU projects
- Directions will depend on customers
New Proof-support extensions

VDMTools

PROSPER proof-engine inside
PROSPER Component View

- VDM-SL Toolbox
- Proof obligation generator
- Translator
- PROSPER proof-engine
- VDM-SL Theories
- VDM-SL Proof support

Front-end

GUI
## Proof obligation generator

![Proof obligations interface](image)

### General PO Information

<table>
<thead>
<tr>
<th>Generated in</th>
<th>At location</th>
<th>Due to</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule</td>
<td>1. 10 c. 11</td>
<td>invariants</td>
<td>New</td>
</tr>
<tr>
<td>Plant</td>
<td>1. 13 c. 33</td>
<td>map application</td>
<td>New</td>
</tr>
<tr>
<td>Plant</td>
<td>1. 0 c. 3</td>
<td>exhaustive function</td>
<td>New</td>
</tr>
<tr>
<td>NumberOfExperts</td>
<td>1. 39 c. 24</td>
<td>map application</td>
<td>New</td>
</tr>
<tr>
<td>ExpertsOnDuty</td>
<td>1. 44 c. 33</td>
<td>invariants</td>
<td>New</td>
</tr>
<tr>
<td>ExpertsOnDuty</td>
<td>1. 44 c. 48</td>
<td>map application</td>
<td>New</td>
</tr>
<tr>
<td>ExpertsOnDuty</td>
<td>1. 42 c. 3</td>
<td>exhaustive function</td>
<td>New</td>
</tr>
<tr>
<td>ExpertToPage</td>
<td>1. 49 c. 0</td>
<td>invariants</td>
<td>New</td>
</tr>
<tr>
<td>ExpertToPage</td>
<td>1. 49 c. 31</td>
<td>map application</td>
<td>New</td>
</tr>
<tr>
<td>ExpertToPage</td>
<td>1. 46 c. 3</td>
<td>satisfiability</td>
<td>New</td>
</tr>
</tbody>
</table>

### Display for Selected Proof Obligations

**Proof Obligation #1:**

1. 10 c. 11: invariants

```plaintext
(forall sch : Schedule a
(forexist exs in set rng sch a
(forexist exs2 in set {} a
(imp Expert(xx2))))
```

**Proof Obligation:**

Schedule, 1. 10 c. 11: invariants

```plaintext
(forexist a in set alarams a
(forexist exs in set dom schedule a
(QualificationOK(schedule(per), a, quali)));
Schedule = map Period to set of Expert
inv sch ==
(forexist exs in set rng sch a

(exs <> {} and
(forexist ex1, ex2 in set exs a
(ex1 <> ex2 <> ex1.expertid <> ex2.expertid)
Period = token;
Expert1 = expertid1 | ExpertId)
```
PROSPER Case studies

- Alarm
- Tracker
- Safer
- Line database (RTRI)
- Interlocking (RTRI)
Development Guidelines for RT

- Use Case Analysis
- VDM-SL System Spec
- Sequential Design Model
- Concurrent Design Model
- Real-Time Design Model
- Implementation
- Host Integration tests
- Target Integration Tests
- Target Validation Tests
- System Acceptance Tests
IFAD and RedVerst processes integration

VDMTool/IFAD modeling and testing process
- executable model designed
- test cases for the model developed
- the model verified

VDM model design iterations

VDM++TesK process
- constraints specified
- test suite designed
- Regression testing infrastructure built

The target software is evolving

Interfaces designed and verified
VDMTools® Tutorial

✓ IFAD Profile
✓ Where does VDM fit in?
✓ VDM++ Overview
✓ Overview of VDMTools®
➢ Demonstration overview
The Cash Dispenser Model

- Model of a system of tills and a central resource.
- Customers interact with tills by inserting a card and entering a PIN.
- Central resources contains detailed records of customers’ bank accounts.
- “Illegal” cards are kept by the till.
A Cash Dispenser Example
There are many tills which can access a central resource containing the detailed records of customers’ bank accounts. A till is used by inserting a card and typing in a PIN (Personal Identification Number) which is encoded by the till and compared with a code stored on the card. After successfully identifying themselves to the system, customers may try to:

1. view the balance of their accounts
2. make a withdrawal of cash
3. ask for a statement of their account to be sent by post.

Information on accounts is held in a central database and may be unavailable. In that case 1) above may not be possible. If the database is available, any amount up to the total in the account may be withdrawn, subject to a fixed daily limit on withdrawals. This means that the amount withdrawn within the day must be stored on the card.

“Illegal” cards are kept by the till.
Development Process

- Analysis (using VDM-SL with API animation)
  - alternative to use cases
  - abstraction from multiple tills
- Design (using Rose VDM++ Link with systematic testing and API animation)
  - abstraction from possible failures of tills
- Implementation (with concurrent VDM++ model and automatic Java code generation combined with user interface)
Further Information

- VDMTools brochures
- Download all VDMTools documentation and executables from [http://www.ifad.dk/Products/VDMTools/executables.htm](http://www.ifad.dk/Products/VDMTools/executables.htm)
- Toolbox Newsletters available at [http://www.ifad.dk/Newsletter/index.htm](http://www.ifad.dk/Newsletter/index.htm)
- Features described at: [http://www.ifad.dk/Products/VDMTools/features.htm](http://www.ifad.dk/Products/VDMTools/features.htm)