Aggressive Model-Driven Development: Synthesizing Systems from Models viewed as Constraints

Tiziana Margaria
Bernhard Steffen
Platforms
(Edward Lee)

Where the Action Has Been:

Giving the red platforms useful modeling properties (e.g. UML, MDA)

Getting from red platforms to blue platforms.
CB Design vs. AMDD

Component Based Design

Component Library

Component Library

M₁  ...  Mn

Compilation/Synthesis

Integration

Integration

Running System

Running System

a board  a chip

AMDD

Model Library

M₁  ...  Mn

Integration as Consistency/Compatibility

Global Model

Synthesis/Technology Mapping

Running System

SoC

T. Margaria, B. Steffen

Monterey 2003
Outline

- AMDD as Actor-Oriented approach
- The ABC as AMDD Environment
- Two Examples
- Conclusions
ABC’s AMDD

Heterogeneous Service Models

Component Model Library

Feature Library

Temporal Constraints and Types

Integration as Consistency/Compatibility

Global SLG

Compilation/Synthesis

Running System

T. Margaria, B. Steffen
Monterey 2003
ABC: Library-based, Actor-oriented Modelling & Design

Service Independent Building Blocks = Actors

Temporal Constraints and Types

Temporal Formulas = Property Specifications

Service Logic Graphs = System Models

Actor Orientation
- model checking composition of behaviors
- separation of behav. interface from implementation

Global SLG

Running System

T. Margaria, B. Steffen
Monterey 2003
ABC: Library-based, Actor-oriented Modelling & Design

- **Automatic** Compilation and Deployment
- Changes, Verification, Updating at the **Modelling Level**
- **Layered** hiding:
  - Distribution,
  - Real Time,
  - Data, …
Outline

- AMDD as Actor-Oriented approach

- The ABC as AMDD Environment

- Two Examples
  - Design
  - Testing
- Conclusions
1) **AMDD for Design:**

The **Online Conference Service**

- **Authors**
- **PC Chair**
- **PC Members**
- **Reviewers**

Flow of information:
- Articles from Authors to PC Chair
- Final Report from PC Chair to Authors
- Report Task from PC Chair to PC Members
- Report from PC Members to PC Chair
- Report Task from PC Members to Reviewers
- Report from Reviewers to PC Members
### Top level Features

- **Roles**
- **Subfeatures**

### List of Articles

The List of Articles page lists all articles you are permitted to read and work with. The legend below the table explains the content of each column.

- **Title**
- **Corresp. Author**
- **Further Authors**
- **Action**
- **Type**

The following table can be sorted according to the following criteria: Article ID, Title, Author. Click on the desired one.

To get an overview of all article abstracts click here: *Article Abstracts*.

To get an overview of all articles delegated to you click here: *Delegated Articles*.

### Table

<table>
<thead>
<tr>
<th>ID</th>
<th>State</th>
<th>Title</th>
<th>Corresp. Author</th>
<th>Further Authors</th>
<th>Action</th>
<th>Type</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>under review</td>
<td>Lecture Notes in Computer Science: Volume Editors’ Instructions</td>
<td>Flowers, Nelly</td>
<td></td>
<td>read article, modify article, remove article, delegate article, mail to author</td>
<td>Technical Paper</td>
<td>001</td>
</tr>
<tr>
<td>002</td>
<td>some reports available</td>
<td>Instructions for Using Author Template</td>
<td>Ferrero, Arnoldo</td>
<td></td>
<td>read article, modify article, remove article, delegate article, mail to author</td>
<td>Technical Paper</td>
<td>002</td>
</tr>
</tbody>
</table>
AMDD for Design

Main Functionalities

Create Newsgroup macro

Macros

SIBs

T. Margaria, B. Steffen
Monterey 2003
Create Newsgroup

The SIB Model

SIBs

<<Name>>
<<Category>>
<<Formal Parameter>>
<<Branches>>
<<Executable Code>>

SIB

CVSCreateConnection

SIB Specification

This SIB reads all required input from the ServiceProperties using the SIB parameters as keys and writes the created CVSSConnection object into the session using the key CVS.connection.
The SIB throws a SIBException if the objects are not available.
If an error occurs the error description will be set to CVS.error in the CallContext.

# param: hostname @ hostname where the server is installed
# param: port @ PortServer port that should be used
# param: rootdir @ rootdir of the CVS repository
# param: username @ username that should be used for the CVS login
# param: password @ password that should be used for the CVS login
# exit: ok if the CVS command was successful
# exit: error if a connection error occurred

SIB
CVSCreateConnection

CLS
CVS

PAR
hostname STR 100 ""
PAR
port STR 100 ""
PAR
rootdir STR 100 ""
PAR
username STR 100 ""
PAR
password STR 100 ""

BR
ok
BR
error

PROP
"java_package" "de.metaframe.evis.application.cvs.sib"
SLG of the Application

Skeleton Service

Features

Global Logic Plane

OCS:
2200 SIBs
3500 branches
Feature Logic

Submit Article
2) Computer/Telephony Integrated Systems

LAN

ISDN Network

Switch

Test Coordinator

Application-PCs

Application-Server
Computer/Telephony Integrated Systems

Intelligent Test Management

Test coordination

ATE context

System under test

HW/SW systems

Controller of device simulator, LAN-Tracer

GUI Test Tool

LAN (CSTA II/III)

Call Center Server

Call Center Client

T. Margaria, B. Steffen

Monterey 2003
The Evolution Problem:

HiPath AllServe Call Center:

- Unified Messaging Services
- Automated Call Distribution (topic experts)
- Interactive Voice Response (IVR) - dynamic teleworking
- ...

30 ACD-groups
8 Supervisors (PCs)
64 Agents (PCs)

T. Margaria, B. Steffen
Monterey 2003
Concrete Test Setting: PCM operation

Test Coordinator

Rational Robot

PCM Application PCs

CSTA II/III

PCM Application Servers
Test Execution

T. Margaria, B. Steffen

Monterey 2003
Conclusions

• Approach to designing and testing role-based, distributed, heterogeneous systems with
  ➢ coordination-oriented model
  ➢ library-based design & test
  ➢ library-based consistency checking
  ➢ incremental formalisation
  ➢ verification-supported design & test

• global, feasible, open, scalable
Go – No Go Conditions

(D. Stidolph – FME 2003)

- Investment curve for the same formal methods and traditional developments
  - Below the Zero Line is Good
  - Negative investment = Profit
- Traditional development breaks even at spec delivery
- Formal methods development costs money until code and unit test is complete
Trend 3: do applications really get more complex faster than our ability to analyze them improves? (no!)

G. Holzmann (FME’03)

Rough indication
- 1968: OS/360 = ~5 Million lines of assembly (~1 Million lines of C)
- 2003: WindowsXP = ~64 Million Lines of C/C++
  (35 years = 23x18 months)
  - INCREASE: \(\sim 2^6\) (64 x)
  - MOORE’s CURVE: \(\sim 2^{23}\) (>8 million x)

Another:
- 1983: trace was 3,507 lines of C
- 2003: spin has 28,687 lines of C
  (20 years = 13.3 x 18 months)
  - INCREASE: \(\sim 2^3\) (8 x)
  - MOORE’s CURVE: \(\sim 2^{13}\) (10,000 x)

Simple observation:
- In the same amount of time
  - 1 Million lines of C code can be compiled and analyzed far more thoroughly today than 1,000 lines of C-equivalent code in 1968
Challenges

- Feature interaction
- Models for legacy systems
- Decoupling of layers
- Couplings in error models
- Mixed qualitative and quantitative aspects
- Incremental methods
- Compilation and Synthesis

Utter specialization of studies:
Will (systems) people (still) be able to talk to each other in 10 years?