COLOSS:
COmposants et LOgiciels SûrS
Safe Components and Softwares

LINA - CNRS - Nantes University

Prague - September, 3-7 2007

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ DE NANTES

Outline

- Team members
- Motivations and Goals
- Research topics and results

  Integration of formal methods / Component specification
- Technical presentation by Pascal

Team presentation

- Creation: july 2005
- Leader: Christian Attiogbé
- Permanent Members:

Pascal André	assistant professor
Gilles Ardourel	assistant professor
Christian Attiogbé	assistant professor
Henri Habrias	Professor

- Associate Member:

Alain Vailly	assistant professor

- PhD Students:

Cédric Stoquer	PhD Student
Mohamed Messabihi	PhD Student
Still looking for another	PhD Student

Background : Object-Orientation and Formal Methods

Motivations and Goals

- Concepts, Techniques and Tools to design correct software
- Main motivations (fundamental challenges):
  - Correct software construction;
  - Software quality and safety;
  - Support for specific development methods;
- Means and method:
  - Use of Multi-faceted formal approaches: specification, reasoning
  - Several steps: specification - verification - refinement
  - At the frontier between theoretical works and their applications.

Z, B, Promela/Spin, PVS, Lotos, Petri nets, Mec/AltaRica, Grafcet, Kmelia, ...

Motivations and Goals

- Concepts, Techniques and Tools to design correct software
- Main motivations (fundamental challenges):
  - Correct software construction;
  - Software quality and safety;
  - Support for specific development methods;
- Means and method:
  - Use of Multi-faceted formal approaches: specification, reasoning
  - Several steps: specification - verification - refinement
  - At the frontier between theoretical works and their applications.

Z, B, Promela/Spin, PVS, Lotos, Petri nets, Mec/AltaRica, Grafcet, Kmelia, ...

Motivations and Goals

- Concepts, Techniques and Tools to design correct software
- Main motivations (fundamental challenges):
  - Correct software construction;
  - Software quality and safety;
  - Support for specific development methods;
- Means and method:
  - Use of Multi-faceted formal approaches: specification, reasoning
  - Several steps: specification - verification - refinement
  - At the frontier between theoretical works and their applications.

Z, B, Promela/Spin, PVS, Lotos, Petri nets, Mec/AltaRica, Grafcet, Kmelia, ...

Multi formalism specifications, multi-faceted analysis

- Motivations: limits of the monoformalism approaches:partial covering of problem and partial analysis
- Formal methods integration
- challenges: decomposition, semantic interoperability, formal analysis
- multi-platforms experiments: B, PVS, Spin, Grafcet, Petri nets

Atacora Platform

Multi formalism specifications, multi-faceted analysis

- Motivations: limits of the monoformalism approaches:partial covering of problem and partial analysis
- Formal methods integration
- challenges: decomposition, semantic interoperability, formal analysis
- multi-platforms experiments: B, PVS, Spin, Grafcet, Petri nets

Atacora Platform

Multi formalism specifications, multi-faceted analysis - Results

- Extension of the B method with the integration of parallel composition operators from process algebra;
- Proposal of multi-faceted analysis method combining theorem proving and model checking with B , SPIN, ProB, Lotos
- Current PhD work on the B/Grafcet interaction, (C. Stoquer);
- Specification method in B of multiprocess systems with dynamic architecture.

ETAPS/FASE'03, QSIC'04, SOFSEM'05
ZB'05, SE'06, ICFEM'06, IEEE-TSE'07

Multi formalism specifications, multi-faceted analysis - Results

- Extension of the B method with the integration of parallel composition operators from process algebra;
- Proposal of multi-faceted analysis method combining theorem proving and model checking with B , SPIN, ProB, Lotos
- Current PhD work on the B/Grafcet interaction, (C. Stoquer);
- Specification method in B of multiprocess systems with dynamic architecture.

ETAPS/FASE'03, QSIC'04, SOFSEM'05
ZB'05, SE'06, ICFEM'06, IEEE-TSE'07

Multi formalism specifications, multi-faceted analysis - Results

- Extension of the B method with the integration of parallel composition operators from process algebra;
- Proposal of multi-faceted analysis method combining theorem proving and model checking with B , SPIN, ProB, Lotos
- Current PhD work on the B/Grafcet interaction, (C. Stoquer);
- Specification method in B of multiprocess systems with dynamic architecture.

ETAPS/FASE'03, QSIC'04, SOFSEM'05
ZB'05, SE'06, ICFEM'06, IEEE-TSE'07

Design and verification of component properties

Developed aspects: modeling, property verification
The motivation: need of models and practical tools to assist users in formal component-based development.

- abstract definition of components and composition
- simple, flexible and expressive
- properties verification: safety, consistency, compatibility...
- from components to code

Kmelia Model - COSTO platform

Design and verification of model properties

Generic verification process for checking UML models consistency (extensible to other models) Managing several verifications because a single property

- can be decomposed into finer ones
- can concern several groups of model elements
- can be verified at different levels of completeness
- can be verified using several techniques with various costs and performances

We designed a generic verification process

- composite verification processes (supports ordering, filtering, results propagation and annotation of faulty elements. . .)
- support for classification of verifications and properties (levels, diagrams...)
- abstracting from the results of different tools and formalisms

Prototype initially supported by a template-based metamodel repository generator.

Design and verification of model properties

Generic verification process for checking UML models consistency (extensible to other models) Managing several verifications because a single property

- can be decomposed into finer ones
- can concern several groups of model elements
- can be verified at different levels of completeness
- can be verified using several techniques with various costs and performances

We designed a generic verification process

- composite verification processes (supports ordering, filtering, results propagation and annotation of faulty elements. . .)
- support for classification of verifications and properties (levels, diagrams...)
- abstracting from the results of different tools and formalisms

Prototype initially supported by a template-based metamodel repository generator.

Design and verification of model properties

Generic verification process for checking UML models consistency (extensible to other models) Managing several verifications because a single property

- can be decomposed into finer ones
- can concern several groups of model elements
- can be verified at different levels of completeness
- can be verified using several techniques with various costs and performances

We designed a generic verification process

- composite verification processes (supports ordering, filtering, results propagation and annotation of faulty elements. . .)
- support for classification of verifications and properties (levels, diagrams...)
- abstracting from the results of different tools and formalisms

Prototype initially supported by a template-based metamodel repository generator.

Component specification and verification with kmelia

- (Kmelia): service-based formal component model
  - services behaviors expressed as extended LTS
  - support for horizontal composition (nested services and behaviors) as well as vertical composition
  - component protocols expressed as services composed of several other ones
- techniques applied on Kmelia models:
  - interface/behavior consistency
  - behavioral compatibility (with Lotos and MEC 4)
  - generation of adaptors to correct some cases of behavioral incompatibility
  - using pre/post conditions to detect inconsistencies in protocols
- COSTO prototype for specifying Kmelia components and analyzing properties
  - support Kmelia model
  - internal verifications and use of existing tools
  - command line, API and eclipse plugins (editors, viewers, wizards for creation, verification...)

WESC'05, OCM/LM0'05, Camode'05, ETAPS/SC'06, SC'07
MOSIM'06, WCAT'06, CAL'06, LMO'07

Component specification and verification with kmelia

- (Kmelia): service-based formal component model
  - services behaviors expressed as extended LTS
  - support for horizontal composition (nested services and behaviors) as well as vertical composition
  - component protocols expressed as services composed of several other ones
- techniques applied on Kmelia models:
  - interface/behavior consistency
  - behavioral compatibility (with Lotos and MEC 4)
  - generation of adaptors to correct some cases of behavioral incompatibility
  - using pre/post conditions to detect inconsistencies in protocols
- COSTO prototype for specifying Kmelia components and analyzing properties
  - support Kmelia model
  - internal verifications and use of existing tools
  - command line, API and eclipse plugins (editors, viewers, wizards for creation, verification...)

WESC'05, OCM/LM0'05, Camode'05, ETAPS/SC'06, SC'07
MOSIM'06, WCAT'06, CAL'06, LMO'07

Component specification and verification with kmelia

- (Kmelia): service-based formal component model
  - services behaviors expressed as extended LTS
  - support for horizontal composition (nested services and behaviors) as well as vertical composition
  - component protocols expressed as services composed of several other ones
- techniques applied on Kmelia models:
  - interface/behavior consistency
  - behavioral compatibility (with Lotos and MEC 4)
  - generation of adaptors to correct some cases of behavioral incompatibility
  - using pre/post conditions to detect inconsistencies in protocols
- COSTO prototype for specifying Kmelia components and analyzing properties
  - support Kmelia model
  - internal verifications and use of existing tools
  - command line, API and eclipse plugins (editors, viewers, wizards for creation, verification...)

WESC'05, OCM/LM0'05, Camode'05, ETAPS/SC'06, SC'07
MOSIM'06, WCAT'06, CAL'06, LMO'07

Component specification and verification with kmelia

- (Kmelia): service-based formal component model
  - services behaviors expressed as extended LTS
  - support for horizontal composition (nested services and behaviors) as well as vertical composition
  - component protocols expressed as services composed of several other ones
- techniques applied on Kmelia models:
  - interface/behavior consistency
  - behavioral compatibility (with Lotos and MEC 4)
  - generation of adaptors to correct some cases of behavioral incompatibility
  - using pre/post conditions to detect inconsistencies in protocols
- COSTO prototype for specifying Kmelia components and analyzing properties
  - support Kmelia model
  - internal verifications and use of existing tools
  - command line, API and eclipse plugins (editors, viewers, wizards for creation, verification...)

WESC'05, OCM/LM0'05, Camode'05, ETAPS/SC'06, SC'07
MOSIM'06, WCAT'06, CAL'06, LMO'07

Perspectives

Ongoing work:

- Extending the Kmelia data and assertion language
- Extending the COSTO Toolbox to deal with consistency using theorem proving
- Using the generic verification process for kmelia verifications
- Mechanizing proof obligations

- Connection with other tools (around Fractal, SOFA, etc)
- Real case studies by joining projects on reverse engineering and related models
- Getting more PhD students (Open PhD position on component and aspect models with the OBASCO Team)

Technical Presentation

Next:
Hierarchical Service Description with Kmelia and Analysis using **COSTO**