Towards Language-Oriented Modeling
Habilitation à Diriger des Recherches
Université de Rennes 1
December 4, 2015

Benoit Combemale (Inria & Univ. Rennes 1)
http://people.irisa.fr/Benoit.Combemale
benoit.combemale@irisa.fr
@bcombemale
Complex Software-Intensive Systems

- Multi-engineering approach
- Some forms of domain-specific modeling
- Software as integration layer
- Openness and dynamicity
Multiple concerns, stakeholders, tools and methods.
Heterogeneous Modeling
"the majority of MDE examples in our study followed domain-specific modeling paradigms"

Model-Driven Engineering (MDE)

"Software languages are software too"

Software Language Engineering (SLE)

• Application of systematic, disciplined, and measurable approaches to the development, deployment, use, and maintenance of software languages

• Supported by various kind of "language workbench"
  • Eclipse EMF, xText, Sirius, GEMOC, Papyrus
  • Jetbrain’s MPS
  • MS DSL Tools
  • Etc.

• Various shapes and ways to implement software languages
  • External, internal or embedded DSLs, Profile, etc.
  • Grammar, metamodel, ontology, etc.

• More and more literature, a dedicated Intl. conference (ACM SLE, cf. http://www.sleconf.org)…
Scientific Context

MDE \text{ metamodeling} \text{ SLE}
Scientific Context

MDE

metamodeling

SLE

Software-Intensive Systems

DSMLs
Scientific Context

- MDE
- SLE
- Software-Intensive Systems
- DSMLs

metamodelling
early dynamic V&V?
Facing the Development of eXecutable DSMLs
Research Statement, Challenges and Contributions

Facing the Development of executable DSMLs

Diagram:
- Abstract Syntax (AS)
- Concrete Syntax (CS)
- Semantics Domain (SD)
- Domain concepts
- concurrency
- domain behaviors
- Dynamic V&V tools

Questions:
1. ?

Keywords:
- Syntax
- Domain
- Behaviors
- Concurrency
- V&V tools
Research Statement, Challenges and Contributions

Facing the Development of Executable DSMLs

xDML Pattern

1 Abstract Syntax (AS)
2 Concrete Syntax (CS)
3 Semantics Domain (SD)
4 Domain behaviors
5 Concurrency
6 Domain concepts
7 Dynamic V&V tools
Facing the Development of executable DSMLs

- xDSML\textsuperscript{1}
  - Abstract Syntax (AS)
  - Concrete Syntax (CS)
  - Semantics Domain (SD)
  - Domain Behavior
  - Domain Concepts
  - Concurrency

xDSML Pattern

Facing the Multiplication of executable DSMLs

- xDSML\textsuperscript{1}
  - Dynamic V&V tools

- xDSML\textsuperscript{2}
Facing the Development of eXecutable DSMLs

Facing the Multiplication of eXecutable DSMLs

xDSML\textsuperscript{1}

Melange
(DSML reuse)

GEMOC Studio
(DSML coordination)

xDSML\textsuperscript{2}

Abstract Syntax (AS)

Concrete Syntax (CS)

Semantics Domain (SD)

xDSML Pattern

structural/axiomatic

domain concepts

corcurrency

Dynamic V&V tools

domain behaviors

behavioral

Research Statement, Challenges and Contributions

Outline

1. Tools and methods for xDSML design and implementation

2. Language globalization for reuse and coordination
THE XDSML PATTERN FOR EXECUTABLE METAMODELING
The xDSML Pattern

- **Execution functions**
  - Internal data/control event (e.g., method call, scheduler) or external (stimuli from env.)

- **Event Definition MetaModel (EDMM)**

- **State Definition MetaModel (SDMM)**

- **Domain Definition MetaModel (DDMM)**

- **Trace Management MetaModel (TM3)**

- **Execution trace**

- **Execution state**

- **Domain model**

Benoît Combemale, Xavier Crégut, Marc Pantel, "A Design Pattern to Build Executable DSMLs and associated V&V tools", In APSEC 2012

The xDSML Pattern

- Mashup of metalanguages (Ecore, OCL, Kermeta) [Jézéquel et al., SoSyM’14]
- Efficient OCL checking [Sun et al., JOT’15]
- Domain-specific execution trace management [Bousse et al., MODELS’14, ECMFA’15]
- Omniscient and multi-dimensional model debugging [Bousse et al., SLE’15]
- Tracing executions back to a xDSML [Combemale et al., ECMFA’11]
Activity Diagram Debugger

https://github.com/gemoc/activitydiagram
Arduino Designer (& Debugger)

- Graphical animation
- Breakpoint definition on model element
- Multi-dimensional and efficient trace management
- Model debugging facilities (incl., timeline, step backward, stimuli management, etc.)

https://github.com/gemoc/arduinomodeling
Arduino Designer (& Debugger)

- Graphical animation
- Breakpoint definition on model element
- Multi-dimensional and efficient trace management
- Model debugging facilities (incl., timeline, step backward, stimuli management, etc.)
- Concurrency simulation and formal analysis

Modern platforms are highly parallel (e.g., many-core, GPGPU, distributed platform).

Complex software systems are highly concurrent systems per se (e.g., IoT, CPS).

https://github.com/gemoc/arduinomodeling

Complex software systems are highly concurrent systems per se (e.g., IoT, CPS).
Reifying Concurrency in xDSML: Limitations

- Concurrency remains implicit and ad-hoc in language design and implementation:
  - Design: implicitly inherited from the meta-language used
  - Implementation: mostly embedded in the underlying execution environment

- The lack of an explicit concurrency specification in language design prevents:
  - leveraging the concurrency concern of a particular domain or platform
  - a complete understanding of the behavioral semantics
  - effective concurrency-aware analysis techniques
  - effective techniques for producing semantic variants
  - analysis of the deployment on parallel architectures
Cross fertilization in languages of the algorithm theory and the concurrency theory

"Concurrency models were generally event-based, and avoided the use of state. They did not easily describe algorithms or the usual way of thinking about them based on the standard model."

Reifying Concurrency in xDSML: Approach

Benoit Combemale, Cécile Hardebolle, Christophe Jacquet, Frédéric Boulanger, Benoit Baudry, "Bridging the Chasm between Executable Metamodelling and Models of Computation," In Software Language Engineering (SLE), 2012.
Reifying Concurrency in xDSML: Approach

The MoCC serves as a (family of) scheduler(s) of the execution functions that manipulate the execution data (i.e. program state).

Benoit Combemale, Cécile Hardebolle, Christophe Jacquet, Frédéric Boulanger, Benoit Baudry, "Bridging the Chasm between Executable Metamodeling and Models of Computation," In Software Language Engineering (SLE), 2012.
Reifying Concurrency in xDSML: Approach

The DSE serve as a **mapping** from the MOC to the DSA

Benoit Combemale, Julien Deantonci, Matias Vara Larsen, Frédéric Mallet, Olivier Barais, Benoit Baudry, Robert France, "Reifying Concurrency for Executable Metamodelling," In Software Language Engineering (SLE), 2013
Reifying Concurrency in xDSML: Approach

The DSEs serve as a protocol between the MOC and the DSA

Florent Latombe, Xavier Crégut, Benoît Combemale, Julien DeAntoni, Marc Pantel, "Weaving concurrency in executable domain-specific modeling languages," In Software Language Engineering (SLE), 2015
Reifying Concurrency in xDSML: Contribution
Reifying Concurrency in xDSML: Contribution
Reifying Concurrency in xDSML: Contribution

Legend
- code generation
- \(<\text{dependsOn}>\)
- \(<\text{conformsTo}>\)

MoC.lib.moc (MoC)
MoC<->DSA.jar
MyDSML DSA-AS.jar
MyDSML .xtend (DSA)
MyDSML .ecore (AS)
Ecore
Kermeta
GEL/ECL
DSE4MyDSML .gel (MoC<->DSA)
MyDSML Concurrency Model.moc

Metamodelling Languages
- (executable) Modeling Language
- (executable) Model

Towards Language-Oriented Modeling – B. Combemale (INRIA and Univ. Rennes 1) – December 4, 2015
The GEMOC Studio

Benoit Combemale, Julien Deantoni, Olivier Barais, Arnaud Blouin, Erwan Bousse, Cédric Brun, Thomas Degueule and Didier Vojtisek, "A Solution to the TTC'15 Model Execution Case Using the GEMOC Studio," In 8th Transformation Tool Contest (TTC), 2015. Overall Winner

http://gemoc.org/studio/
Coping with Semantic Variation Points

Florent Latombe, Xavier Crégut, Julien Deantoni, Marc Pantel, Benoit Combemale, "Coping with Semantic Variation Points in Domain-Specific Modeling Languages", In EXE@MoDELS 2015.
"A clear challenge, then, is how to integrate multiple DSLs."

Multiplication of DSMLs

Increasing number of application domains

Increasing number of stakeholders and concerns
ON THE GLOBALIZATION OF MODELING LANGUAGES
Globalization of Modeling Languages

- DSMLs are developed in an independent manner to meet the specific needs of domain experts,

- DSMLs should also have an associated framework that regulates interactions needed to support collaboration and work coordination across different system domains.

Globalization of Modeling Languages

Supporting coordinated use of modeling languages leads to what we call the globalization of modeling languages, that is, the use of multiple modeling languages to support coordinated development of diverse aspects of a system.
Globalization of Modeling Language

- Context: new emerging DSML in open world
  ⇒ impossible a priori unification
  ⇒ require a posteriori globalization

- Objective: socio-technical coordination to support interactions across different system aspects
  ⇒ Language-based support for technical integration of multiples domains
  ⇒ Language-based support for social translucence

- Community: the GEMOC initiative (cf. http://gemoc.org)

Towards Language Interfaces

- A language interface is a contract that exhibits the relevant information for a given purpose (i.e., to support specific composition operators)
MELANGE: MODULAR AND REUSABLE DSML DESIGN AND IMPLEMENTATION
Jim Steel, Jean-Marc Jézéquel, "On model typing," In Software and System Modeling (SoSyM), 2007

- Rich set of subtyping relationships [Guy et al., ECMFA'12]
- Support of OCL in subtyping relationships [Sun et al., ECMFA'13]
Reuse is not enough! Context matters!

Imported / inherited DSMLs may not fit exactly domain-specific requirements

→ Finely tune with customization facilities
Modular and Reusable xDSML Development

Thomas Degueule, Benoît Combemale, Arnaud Blouin, Olivier Barais, Jean-Marc Jézéquel, "Melange: a meta-language for modular and reusable development of DSLs," In Software Language Engineering (SLE), 2015
A language-based, model-oriented programming language

http://melange-lang.org
CLOSED WORLD

Variability model

Language derivation

Variability-based development model for DSLs
- Variability modeling
- Components-based languages development

FAMILIES OF LANGUAGES

Variants

OPEN WORLD

Language Manipulation
- Evolution
- Extension
- Restriction
- Customization
- Assembly

Typing Theory for Agile Modeling
- Language interfaces
- Model polymorphism
- Viewpoints management
B-COOL: BEHAVIORAL MODEL COORDINATION
Towards a behavioral coordination of xDSMLs

- The application of the MOC to a given model results in an event structure.

- Consequently, the MOC defines a symbolic event structure, whose some events are mapped to the DSA (i.e., the visible model state changes).

⇒ This mapping can serve as a behavioral language interface used to define patterns that coordinate conforming models.

Matias Ezequiel Vara Larsen, Julien Deantoni, Benoit Combemale, Frédéric Mallet, "A Behavioral Coordination Operator Language (BCOoL)," In MODELS 2015
Behavioral Coordination Operator

- Data
- Control
- Communication

Event-driven behavioral interface

Concurrent execution of *homogeneous* domain-specific models

- Data
- Control
- Communication
Behavioral Coordination Operator

Concurrent execution of heterogeneous domain-specific models

Event-driven behavioral interface

- Data
- Control
- Communication

xDsML

Semantics

MoCC

DSA

AS

DSE

(Domain Specific Event)

xDsML'

DSE

AS

DSA

MoCC

• Data
• Control
• Communication

Concurrent execution of heterogeneous domain-specific models
Behavioral Coordination Operator

Matias Ezequiel Vara Larsen, Julien Deantoni, Benoit Combemale, Frédéric Mallet, "A Behavioral Coordination Operator Language (BCOoL)," In MODELS 2015
Behavioral Model Coordination

Executable Language 1

Behavioral Type (GEL)

Behavioral Interface

Structural Interface

Behavioral Coordination Patterns

Language Coordination Operators (BCOoL)

Executable Language 2

Behavioral Type (GEL)

Behavioral Interface

Structural Interface

e.g., control flow

<<Conforms To>>

Generates

Symbolic Event Structure (CCSL)

Model Behavioral Interface

Model Structural Interface

Heterogeneous Coordination Model

Data Structure (EMF)

Model Behavioral Interface

Model Structural Interface

Data Structure (EMF)

<<Conforms To>>

Generates

Symbolic Event Structure (CCSL)

Model Behavioral Interface

Model Structural Interface

Data Structure (EMF)

execute

execute

execute

Generic Execution Engine (Parametrized by Language 1)

Generic Execution Engine (Parametrized by Language 2)

Coordination Engine (Parametrized by the Coordination Model)
Behavioral Model Coordination
CONCLUSION AND PERSPECTIVES
Conclusion

Contributions to the support of a language-oriented modeling with

tools and methods to implement, reuse and coordinate xDSMLs and associated V&V tools
Major breakthroughs

• Modular definition of xDSMLs

• Reification of the concurrency in xDSMLs

• Language interfaces for reuse and behavioral coordination
Perspectives


Metamorphic DSMLs for fitting user needs

Model Experiencing Environments for informed decision and broader engagement in smart technologies

DSMLs as a key pivot for the socio-technical coordination

Approximate DSML runtimes for design space exploration, runtime adaptation and security

B. Combemale, B. Cheng, A. Moreira, J.-M. Bruel, J. Gray, "Modeling for Sustainability," SEIS@ICSE 2015 (submitted)
"If you believe that language design can significantly affect the quality of software systems, then it should follow that language design can also affect the quality of energy systems. And if the quality of such energy systems will, in turn, affect the livability of our planet, then it’s critical that the language development community give modeling languages the attention they deserve."

– Bret Victor (Nov., 2015), http://worrydream.com/ClimateChange
Thank you
DiverSE

Functional diversity

Language diversity

Execution diversity

Failure diversity

Model-driven engineering
Scientific collaborations
In memory of Prof. Robert B. France,
A mentor, an academic father and far beyond.
Rest in peace, Robert.

http://people.irisa.fr/Benoit.Combemale/tribute-robert-france
TIME TO DRINK CHAMPAGNE and dance ON THE table

ILE de SEIN ENEZ-SUN