Towards Language-Oriented Modeling

Benoit Combemale (Univ. Rennes 1, IRISA, DiverSE team)

http://people.irisa.fr/Benoit.Combemale
benoit.combemale@irisa.fr
@bcombemale
Complex Software-Intensive Systems

- Multi-engineering approach
- Domain-specific modeling
- High variability and customization
- Software as integration layer
- Openness and dynamicity
Multiple Concerns

- Aerodynamics
- Avionics
- Mechanical Structure
- Propulsion System
- Airlines
- Human-Machine Interaction
- Environmental Impact
- Safety Regulations
- Communications
- Navigation
- Authorities
Heterogeneous Modeling
"Perhaps surprisingly, the majority of MDE examples in our study followed domain-specific modeling paradigms"

Domain-Specific Languages (DSLs)

• Tailored for a particular kind of problem, with dedicated notations (textual or graphical), support (editor, checkers, etc.)
• Promises: more « efficient » languages for resolving a set of specific problems in a domain

• DSL ~ Modeling Language!
Another lesson we should have learned from the recent past is that the development of 'richer' or 'more powerful' programming languages was a mistake in the sense that these baroque monstrosities, these conglomerations of idiosyncrasies, are really unmanageable, both mechanically and mentally.

I see a great future for very systematic and very modest programming languages

ACM Turing Lecture, « The Humble Programmer »
Edsger W. Dijkstra 1972
Empirical Assessment of MDE in Industry

John Hutchinson, Jon Whittle, Mark Rouncefield
School of Computing and Communications
Lancaster University, UK
+44 1524 510492
{j.hutchinson, j.n.whittle, m.rouncefield}@lancaster.ac.uk

Steinar Kristoffersen
Østfold University College and Møreforskning Molde AS
NO-1757 Halden
Norway
+47 6921 5000
steinar.kristoffersen@hiof.no

Model-Driven Engineering Practices in Industry

John Hutchinson
School of Computing and Communications
Lancaster University, UK
+44 1524 510492
{j.hutchinson@lancaster.ac.uk}

Mark Rouncefield
School of Computing and Communications
Lancaster University, UK
+44 1524 510492
{m.rouncefield@lancaster.ac.uk}

Jon Whittle
School of Computing and Communications
Lancaster University, UK
+44 1524 510492
{j.n.whittle@lancaster.ac.uk}

« Domain-specific languages are far more prevalent than anticipated »

2011

Editors (textuals, graphicals, …)

Documentation generators

Test generators

Simulators

Analyzers

Refactoring

Checkers (static & dynamics)

Compilers

Translators

Code generators

Etc.
"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

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

- Supported by various kind of "language workbench"
  - Eclipse EMF, Xtext, Sirius, Melange, GEMOC, Papyrus
  - Jetbrain’s MPS
  - Spoofax
  - MS DSL Tools
  - Etc.

The Kermeta Workbench (since 2005)

- **Modular design** of DSLs
  - One meta-language per language concern (merge/weave)
    - Ecore, OCL, Xtend
    - But also: QVTo, fUML, Alf, Ket, Xsd…
  - Static introduction mechanism (aspect)

- **Provides a model oriented action language** to support common model manipulation tasks
  - to implement (E)Operation’s bodies
  - Imperative, statically typed, object-oriented, aspect-oriented (aspect/context, require), model-oriented, DbC, Unit testing
  - Java and Xtend compliant, and based on EMF
  - Run as Eclipse plugin or as standard Java application

- **Efficient implementation** of DSMLs
  - Mashup of the meta-languages to efficient bytecode
  - Integrated with third-party tools (EMF compliant)

Breathe Life Into Your Designer!

Towards Language-Oriented Modeling – B. Combemale (Univ. Rennes 1) – May 3rd, 2017
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/
Activity Diagram Debugger

https://github.com/gemoc/activitydiagram
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
Reifying Concurrency in xDSML: Approach

Abstract Syntax (AS) → Concrete Syntax (CS)

Abstract Syntax (AS) → Semantic Domain (SD)

Concrete Syntax (CS) → Concrete Syntax (CS)

Domain-Specific Actions

Model of Concurrency

Algorithm Theory

Concurrency Theory
Weave Concurrency Constraints Into Your DSL!

Benoit Combemale, Julien Deantoni, Matias Vara Larsen, Frédéric Mallet, Olivier Barais, Benoit Baudry, Robert France, "Reifying Concurrency for Executable Metamodelling," In Software Language Engineering (SLE), 2013
Activity Diagram Debugger

Benoit Combemale, Julien Deantonii, Matias Vara Larsen, Frédéric Mallet, Olivier Barais, Benoit Baudry, Robert France, "Reifying Concurrency for Executable Metamodelling," In Software Language Engineering (SLE), 2013
Florent Latombe, Xavier Crégut, Julien Deantoni, Marc Pantel, Benoit Combemale, "Coping with Semantic Variation Points in Domain-Specific Modeling Languages", In EXE@MoDELS 2015.
Arduino Designer (& Debugger)

https://github.com/gemoc/arduinomodeling
Transformation Lg Debugger

https://github.com/tetrabox/minitl

TETRABox
http://modeltransformation.net/tetrabox/
Wimmer, Bousse et al.
Farming System Modeling

https://github.com/gemoc/farmingmodeling
"A clear challenge, then, is how to integrate multiple DSLs."

Multiplication of DSMLs

Increasing number of application domains

Increasing number of stakeholders and concerns
Towards Language Interfaces

A language interface is a contract that exhibits the relevant information from a language specification for a given purpose (i.e., to support specific composition operators)

Model type
(for language reuse)

Thomas Degueule, Benoît Combemale, Arnaud Blouin, Olivier Barais, Jean-Marc Jézéquel, "Melange: a meta-language for modular and reusable development of DSLs," In SLE 2015

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

Symbolic event structure
(for behavioral coordination)

"On Language Interfaces," Thomas Degueule, Benoit Combemale, Jean-Marc Jézéquel. PAUSE: Present And Ulterior Software Engineering, 2017. Cf. https://hal.inria.fr/hal-01424909v1
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)

The GEMOC Initiative

An open and international initiative to
• coordinate (between members)
• disseminate (on behalf the members)
worldwide R&D efforts
on the globalization of modeling languages

http://gemoc.org
@gemocinitiative
The GEMOC Community
The GEMOC Studio

Design and compose your executable DSMLs

Modeling Workbench

Edit and debug your heterogeneous models

http://gemoc.org/studio
Conclusion

• Separation of Concerns and Abstractions are the key enablers of Software Engineering
  • *Modular design* through multiple models
  • *Abstract design* though domain-specific models

• From MDE to SLE
  • Language workbenches support DS(M)L development
    ➢ *towards language-oriented modeling*

• On the globalization of modeling languages
  • Integration of the various heterogeneous models
  • Language interfaces to support structural and behavioral relationships between domains (i.e., DSLs)
    ➢ *DSMLs to support the socio-technical coordination*