

Fourth Workshop on the Globalization of Modeling Languages (GEMOC 2016) *

Julien Deantoni¹, Eugene Syriani²,
Moussa Amrani³, Alexander Knapp⁴,
Jeff Gray⁵

¹ University of Côte d'Azur, I3S lab, INRIA Aoste, France

² University of Montreal, Canada

³ University of Namur, Belgium

⁴ Universität Augsburg, Germany

⁵ University of Alabama, USA

Abstract. The fourth edition of the GEMOC workshop was co-located with the MODELS 2016 conference in Saint-Malo, France. Once again, the workshop provided an open forum for sharing experiences, problems and solutions related to the challenges of using multiple modeling languages in the development of complex software-based systems. During the workshop, both well matured and preliminary approaches were presented and discussed, ideas and opinions exchanged, and constructive feedback provided to authors of accepted papers. A major objective was to encourage more collaborations and to consolidate the community that focuses on providing solutions supporting the globalization of domain-specific modeling languages, *i.e.*, the coordinated use of multiple languages throughout the development of complex systems. This report summarizes the presentations and discussions that took place in the 2016 edition of the workshop.

1 Introduction

To cope with complexity, modern software-intensive systems are often separated into different concerns to serve the needs of diverse stakeholders. These different concerns are often associated with specialized description languages and technologies that are based on concern-specific problems and solution concepts. Software developers are thus faced with the challenging task of integrating the different languages and associated technologies used to produce software artifacts in the different concern spaces.

GEMOC 2016 was a full-day workshop that brought together researchers and practitioners in the modeling languages community to discuss the challenges associated with integrating multiple, heterogeneous modeling languages. The languages of interest range from requirements to design and runtime languages, and include both general-purpose and domain-specific languages. Challenges related to engineering composable languages, well-formed semantic composition of languages and reasoning about systems described using heterogeneous languages were of particular interest. Following

* This workshop is supported by the GEMOC initiative (<http://gemoc.org>).

the three previous editions, a major objective was to continue collaborations and to expand a community that is focused on solving the problems arising from the globalization of modeling languages; *i.e.*, the use of multiple domain-specific languages (DSL) to support coordinated development of diverse aspects of a system.

The GEMOC 2016 workshop was co-located with MODELS 2016 in Saint-Malo, France, on October 4th, 2016. In this report, we document the various presentations, as well as the enthusiastic and intense discussions that took place during the workshop. The workshop report is organized as follows. Section 2 gives a broad overview of the workshop, including the topics of interest and relevant application domains. Section 3 describes the paper review and selection process, as well as the structure of the workshop. Section 4 summarizes the presentations made in the first half of the workshop day. Section 5 then presents the main points raised in the discussions that followed the presentations. The intent of the discussions was to discuss specific points, decided by the workshop participants, related to the globalization of modeling languages. The results of these discussions are summarized in Section 5. Conclusions are drawn in Section 6.

2 Workshop Overview

Software-intensive systems are becoming more complex, driven by the need to integrate concerns across multiple domains. Consequently, the development of such systems requires the integration of many different concerns and skills. These concerns are usually covered by different languages, with specific concepts, technologies and abstraction levels. This permutation of multiple languages eases the development related to a combination of specific concerns, but raises language and technology integration problems at the different stages of the software life cycle. In order to reason about the global system, it becomes necessary to explicitly describe the different kinds of relationships that exist between the various languages used in the development of a complex system. To support effective language integration, there is a pressing need to reify and classify these relationships, as well as the language interactions that the relationships enable.

In this context, the workshop attracted submissions that outline language integration approaches and case studies that identify and discuss well-defined problems about the management of relationships between heterogeneous modeling languages. These presentations catalyzed deep discussions among the participants with the goal of identifying hot topics of interest for the later discussions.

The Call for Papers explicitly solicited contributions that described a vision for developing the necessary breakthroughs in software languages to support global software engineering, *i.e.*, breakthroughs that lead to effective technologies supporting different forms of language integration, including language collaboration, interoperability and composability, as well as the related social coordination between developers that use different languages.

The topics of interest for GEMOC 2016 were:

- Composability and interoperability of heterogeneous modeling languages
- Language integration challenges, from requirement to design, for analysis and simulation, during runtime, etc.
- Model and metamodel composition

- Language interface, viewpoint
- Heterogeneous modeling and simulation
- Multi-paradigm modeling
- Language-based socio-technical coordination

Submissions describing practical and industrial experience related to the use of heterogeneous modeling languages were also encouraged, particularly in the following application domains:

- Cyber-Physical Systems, System of Systems
- Internet of Services, Internet of Things
- Complex Adaptive Systems
- Smart City, Smart Building, Home automation

3 Workshop Organization

Julien Deantoni, Jeff Gray, and Eugene Syriani organized and chaired the program committee (PC) for this fourth GEMOC edition. The workshop website⁶ and the call for papers (CfP) were online, about 6 months before the workshop date. Apart from the workshop website, the CfP was announced on different professional mailing lists (*e.g.*, SEWORLD, DBWorld, planetmde).

Each submission was reviewed by at least three members of the PC. Papers were selected based on their relevance to the topics for the workshop and the reviews provided by PC members. The organizers are very grateful to the PC members for performing this important service to the community and for the quality of their work. The PC consisted of: Cédric Brun (OBEO, France), Marsha Chechik (University of Toronto, Canada), Tony Clark (Middlesex University, United Kingdom), Benoit Combemale (INRIA and University of Rennes 1, France), Jonathan Corley (University West Georgia, USA), Julien Deantoni (University of Côte d’azur, France (chair)), Jeff Gray (University of Alabama, USA (chair)), Jean-Marc Jézéquel (University of Rennes 1, France), Ralf Laemmel (Universität Koblenz-Landau, Germany), Marjan Mernik (University of Maribor, Slovenia), Gunter Mussbacher (McGill University, Canada), Florian Noyrit (CEA List, France), Richard Paige (University of York, United Kingdom), Bernhard Rumpe (RWTH Aachen University, Germany), Matthias Schöttle (McGill, Canada), Eugene Syriani (University of Montreal, Canada) Mark Van Den Brand (Eindhoven University of Technology, The Netherlands), Hans Vangheluwe (University of Antwerp, Belgium), Eric Van Wyk (University of Minnesota, USA), and Markus Voelter (Independent, Germany).

The format of the workshop reflected its goals: to provide constructive feedback on submitted papers and models on the coordinated use of different modeling languages, and to foster collaborations and community building. The workshop day was split into three parts. The first part started with a quick introduction about the major achievements of the GEMOC initiative during the year and was followed by a keynote on “Live Modeling” given by Tijs van der Storm. Then, a session with presentations of the accepted

⁶ Cf. <http://gemoc.org/gemoc2016>.

papers took place. The third part of the day was dedicated to open discussions whose topics were chosen by the workshop participants. We unfortunately did not have enough time to address all the topics identified as important by the participants. This workshop report is the result of the beneficial discussions that occurred throughout the workshop.

4 Paper Summaries

The papers and some of the associated talks are available on the workshop website (<http://www.gemoc.org/gemoc2016>). Below is a summary of each presented paper.

[1] *Embedding Component Behavior DSLs into the MontiArcAutomaton ADL* (Arvid Butting, Bernhard Rumpe and Andreas Wortmann) This paper presents a DSL dedicated to the injection of application-specific behaviors (behavioral DSL) into component and connector ADLs. This DSL enables language integrators to easily embed DSLs as required by contributing domain experts. The goal is to provide executability of the resulting integrated language. The integration relies on the notion of placeholders that the domain experts have to extend in order to integrate their specific behavior. The integration is then specified externally and the result is a new language whose syntax and compilers are integrated according to the explicit specification. The goal of the integration is to obtain a language equipped with a translation semantics resulting from the integration of each translational semantics.

[2] *Alignment of Viewpoint Heterogeneous Design Models: “Emergency Department” Case Study* (Mahmoud El Hamlaoui, Bernard Coulette, Sophie Ebersold, Saloua Benani, Mahmoud Nassar, Adil Anwar, Antoine Beugnard, Yassine Jamoussi and Hanh-Nhi Tran) This paper identifies and considers problems arising when a system is described by different heterogeneous multi-views; more precisely, when a consistent representation is required across these different views. The authors propose to rely on specific correspondences between the different views, with the corespondences stored in a virtual global model called MIC (Model of correspondences between models), which conforms to a uniform language based on a MetaModel of Correspondences. The paper highlights, based on a case study, how these correspondences can be used to derive properties according to the semantics of the correspondence.

[3] *Towards Semantic Integration of Plant Behavior Models with AutomationML’s Intermediate Modeling Layer* (Tanja Mayerhofer, Manuel Wimmer, Luca Berardinelli, Emanuel Mätzler and Nicole Schmidt) This paper is focused on the emerging IEC standard for storing and exchanging engineering data among the heterogeneous software tools involved in the engineering of production systems. The paper explores the formalization of the standard pivot language in order to later allow the semantically correct composition of the different behavioral modeling languages currently employed in the production domain (e.g., Gantt charts, scenarios, impulse diagrams). The formalization of the pivot language is given by a metamodel (to define the syntax) and its operational semantics (to define its behavioral semantics). The result is implemented in a tool and several case studies are presented. The pivot language may then be used as the semantic domain for the composition of heterogeneous plant behavior models.

[4] *fSysML: Foundational Executable SysML for Cyber-Physical System Modeling* (Omar Badreddin, Vahdat Abdelzad, Timothy Lethbridge and Maged Elaasar) This paper presents preliminary work on a language the authors name fSysML. The main contribution is the definition of a textual syntax of SysML together with an extension incorporating goal modeling. The paper shows an example of the use of fSysML in the development of a self-adaptive system from the healthcare domain. Most cyber-physical systems interact with the physical world (by definition); however, the way it is managed in fSysML is not detailed in this paper.

[5] *A New Approach of Qualitative Simulation for the Validation of Hybrid Systems* (Slim Medimegh, Jean-Yves Pierron, Jean-Pierre Gallois and Frédéric Boulanger) This paper presents a new approach for the qualitative simulation of hybrid systems, which relies on an abstract model of the state variables and of the qualitative aspects of their evolution. Just like abstract interpretation, which provides domains for values instead of enumerating them, the proposed approach enables abstraction over the derivatives of the continuous equations instead of solving them. This is an interesting first-step toward the use of physical models at a (very) high abstraction level, when the lack of precision about some parameters forbids *usual* simulations.

5 Discussions

The discussions were split into two parts. The first discussion was dedicated to the elicitation of topics of importance for the participants. Many topics were proposed, such as:

- Physical modeling in the globalization process. What are the differences and commonalities?
- How to help sociological globalization between different disciplines (e.g., physicists)?
- How experiments (about proper interfaces) should be driven to help understanding globalization?
- How can the role/nature/purpose of a model impact its interface (extension, refinement, etc). Can the usage of the interface also impact its content?
- What are the potential links between language interfaces and model interfaces?
- Can we use a megamodel to “classify” the correspondances between models?

Given the limited timeframe, we were not able to cover all of these interesting topics and associated questions. In what follows, we summarize the discussions around some of these topics.

Notions of Language Interfaces

Most of the discussions were concentrated around the notion of *Language Interface*. Participants of the workshop first started to enumerate language interfaces that already exist in popular heterogeneous modeling approaches like Monticore [6], Modhel’X [7]

or Ptolemy [8], but then the discussion broadened to take into consideration what is already existing in programming languages and on the web.

Many, if not all, General-Purpose Programming Languages possess the ability to execute pieces of code written in another language, thus favoring language interaction during execution. For example, it is possible to call a C procedure (*e.g.*, a driver routine already written for a particular device) from Java. There is the same kind of approach in web service-based approaches. This acts like a *connector* between models written in various languages, more than a fully-fledged interface. However, this helps in understanding that this kind of language interface is dedicated to the *co-execution* of different models written in different languages. The underlying implicit language interface is very similar to other frameworks dedicated to the same purpose.

For example, Ptolemy, Modhel'X, as well as classical general-purpose connectors like JNA contain the same key ingredients in their (unfortunately implicit) interface. To use such connecting capabilities, one needs to know how to

- convert values from the calling language into the callee's value format, thus conforming to the type system of the target language;
- know the proper format / syntax of encapsulated code snippets to be called (be it a procedure, method, etc.), which is usually referred to as the *signature*;
- be aware of the behavioral semantics of the languages: *e.g.*, knowing if the method calls are synchronous or asynchronous, understanding how the notion of time is considered, or gaining insight into how the concurrency model and other different parts of the models are scheduled.

This emphasized the fact that the purpose of the language interface is of great importance for its definition. For instance, the kind of language interface dedicated to model coordination ([9]) is different from language interfaces dedicated to omniscient debugging ([10]), which is also different from a language interface dedicated to language composition like in Monticore [6] or Melange [11]. Of course, there is no “one size fits all”; for instance, the internal observability may vary according to the purpose of the language and its use by domain experts. There are a variety of language interfaces that must be classified according to their purpose and may benefit from a clear taxonomy.

At the end of the discussion, two different usages of language interfaces emerged. On the one hand, there are language interfaces that are exploited at the language level to understand the “possibilities/characteristics” of the models that conform to these languages (typically language interfaces dedicated to model co-executions). On the other hand, there are language interfaces that are used to manipulate the languages themselves, both syntactically and semantically (typically language interfaces dedicated to language composition).

Finally, it was also pointed out that the globalization problem and its impact on language interfaces may depend on whether the languages were done without the globalization preoccupation in mind (*a posteriori* globalization) or whether the languages were developed with the globalization concerns as a core design objective (*a priori* globalization).

Differences and Commonalities Between Software Globalization and Cyber-Physical Globalization

Nowadays, and typically during this workshop, most of the papers focused on the globalization of *Software* modeling languages. However, there is a need to better communicate between computer scientist and other scientist (*e.g.*, mechanical engineers) to manage the globalization of modeling languages in Cyber-Physical Systems. One proposition made during the workshop was to use a language interface, which enables sociological translucence and from which it would be possible to generate specific views, dedicated to the discussion between scientists from different domains.

It is clear that, for software languages and for co-execution, it is mandatory to have a view on the behavioral semantics of the language. An interesting point is that this is not true for physical models. Usually, what is exposed from physical languages is more focused on the solver than the semantics of the language. Perhaps this is due to a homogeneous way of specifying physical behavior using differential equations. If this is the case, then the heterogeneity comes only from the way of approximating the equation to provide a simulation. The Functional Mockup Interface (FMI⁷) provides a simple interface exposing a specific part from the AST of the language (*e.g.*, input and output variables) and services the underlying solver can provide or be asked to render.

It is not clear from the literature and our recent discussions how this difference between cyber and physical language interfaces can be better understood/integrated/exploited to ease the globalization between each other.

6 Conclusions

The fourth edition of the workshop met its goals and thus can be considered a success. The discussions that took place during the workshop were of very high quality and provided insights into some of the pressing problems associated with the use of multiple heterogeneous languages. A key result is the strong participation and involvement of the attendees. The ongoing research that was reported in the workshop and the discussions that took place are a good indication that the community around these problems is strengthened. For more information about the GEMOC initiative, please visit the following website: <http://gemoc.org>.

Acknowledgments

GEMOC 2016 was supported by the GEMOC initiative (cf. <http://gemoc.org>), which promotes research that seeks to develop the necessary breakthroughs in software languages to support global software engineering, *i.e.*, breakthroughs that lead to effective technologies supporting different forms of language integration, including language collaboration, interoperability and composability.

We want to thank all of the workshop authors, all members of the program committee for their reviews, and all participants (new and past participants) who help make the community active!

⁷ <http://fmi-standard.org/>

References

1. Butting, A., Rumpe, B., Wortmann, A.: Embedding Component Behavior DSLs into the MontiArcAutomaton ADL. In: GEMOC workshop 2016 - International Workshop on The Globalization of Modeling Languages, Saint Malo, France (2016) http://gemoc.org/wp-content/uploads/2016/03/GEMOC_2016_paper_1.pdf.
2. El Hamlaoui, M., Coulette, B., Ebersold, S., Bennani, S., Nassar, M., Anwar, A., Beugnard, A., Yassine, J., Tran, H.N.: Alignment of viewpoint heterogeneous design models: “Emergency Department” Case Study. In: GEMOC workshop 2016 - International Workshop on The Globalization of Modeling Languages, Saint Malo, France (2016) http://gemoc.org/wp-content/uploads/2016/03/GEMOC_2016_paper_2.pdf.
3. Mayerhofer, T., Wimmer, M., Berardinelli, L., Mätzler, E., Schmidt, N.: Towards Semantic Integration of Plant Behavior Models with AutomationML’s Intermediate Modeling Layer. In: GEMOC workshop 2016 - International Workshop on The Globalization of Modeling Languages, Saint Malo, France (2016) http://gemoc.org/wp-content/uploads/2016/03/GEMOC_2016_paper_5.pdf.
4. Badreddin, O., Abdelzad, V., Lethbridge, T., Elaasar, M.: fSysML: Foundational Executable SysML for Cyber-Physical System Modeling. In: GEMOC workshop 2016 - International Workshop on The Globalization of Modeling Languages, Saint Malo, France (2016) http://gemoc.org/wp-content/uploads/2016/03/GEMOC_2016_paper_3.pdf.
5. Medimegh, S., Pierron, J.Y., Gallois, J.P., Boulanger, F.: A New Approach of Qualitative Simulation for the Validation of Hybrid Systems. In: GEMOC workshop 2016 - International Workshop on The Globalization of Modeling Languages, Saint Malo, France (2016) http://gemoc.org/wp-content/uploads/2016/03/GEMOC_2016_paper_4.pdf.
6. Krahn, H., Rumpe, B., Völkel, S.: Monticore: a framework for compositional development of domain specific languages. *International journal on software tools for technology transfer* **12**(5) (2010) 353–372
7. Hardebolle, C., Boulanger, F.: Modhel’x: A component-oriented approach to multi-formalism modeling. In: *International Conference on Model Driven Engineering Languages and Systems*, Springer (2007) 247–258
8. Buck, J.T., Ha, S., Lee, E.A., Messerschmitt, D.G.: Ptolemy: A framework for simulating and prototyping heterogeneous systems. (1994)
9. Vara Larsen, M.E., Deantoni, J., Combemale, B., Mallet, F.: A Behavioral Coordination Operator Language (BCOoL). In Lethbridge, T., Cabot, J., Egyed, A., eds.: *International Conference on Model Driven Engineering Languages and Systems (MODELS)*. Number 18, Ottawa, Canada, ACM (September 2015) 462 to be published in the proceedings of the Models 2015 conference.
10. Bousse, E., Corley, J., Combemale, B., Gray, J., Baudry, B.: Supporting efficient and advanced omniscient debugging for xdsmpls. In: *Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering*, ACM (2015) 137–148
11. Degueule, T., Combemale, B., Blouin, A., Barais, O., Jézéquel, J.M.: Melange: A meta-language for modular and reusable development of dsls. In: *Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering*, ACM (2015) 25–36