Second Workshop on the Globalization of Modeling Languages (GEMOC 2014)

Workshop Report

Benoit Combemale[1], Julien De Antoni[2], Robert B. France[3], and David Méndez-Acuña[1]

[1] University of Rennes 1 / Inria, France
[2] University of Nice Sophia Antipolis, France
[3] Colorado State University, USA

1 Introduction

The development of modern complex software-intensive systems involve various concerns, which serve diverse stakeholders. These different concerns are often associated with specialized description languages and technologies, which are based on concern-specific problems and solution concepts. Software and systems engineers 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.

Part of the series, GEMOC 2014 was a full-day workshop that brought together researchers and practitioners in the modeling community to discuss the challenges associated with integrating multiple, heterogeneous modeling languages. The languages of interest range from requirements to runtime languages, and include both general-purpose and domain-specific languages. Challenges related to engineering composable languages, well-formed semantic composition of languages and to reasoning about systems described using heterogeneous languages were discussed.

The workshop GEMOC 2014 was co-located with MODELS 2014 in Valencia, Spain, on September 28th, 2014. 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 organization of the workshop. Section 4 summarizes the presentations made in the first half of the workshop day. Section 5 then presents a summary of the results of that discussion that had place during the last part of the workshop.

2 Workshop Overview

Software intensive systems are becoming more and more complex and the range of the application domains is constantly growing. 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 multiplication of languages eases the development of a specific concern 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 the relationships that can exist between the different languages used in the development of a complex system. To support effective language integration, there is a pressing need to rectify and classify these relationships, as well as the language interactions that the relationships enable.

In this context, the workshop GEMOC 2014 attracted submissions that include outlines of language integration approaches, case studies, or that identify and discuss well defined problems about the management of relationships between heterogeneous modeling languages. The goal is to facilitate good discussions among the participants that lead to an initial classification of the kinds of language relationships and their management.

The call for papers explicitly solicited contributions, descriptions of case studies on coordinated use of multiple modeling languages, and/or descriptions of practical experiences, opinions and related approaches. Authors were be invited to submit papers describing (i) their language integration experience, or (ii) novel approaches for integrating modeling languages. Authors were also be invited to store full versions of models used to illustrate their novel approach or experience in the Repository for Model Driven Development (ReMoDD). This allows to share the models with participants and the wider modeling community before and after the workshop.

The topics of interest for GEMOC 2014 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
- Heterogeneous modeling and simulation

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,
- Smart City, Smart Building, Home automation,
- Complex Adaptive Systems,
- Internet of Services, Internet of Things.

3 Workshop Organization

The format of the workshop reflected the goals of the workshop: constructive feedback on submitted papers on the conjoint use of different modeling languages, collaboration, and community building. The format of the workshop

was the one of a working meeting. Hence, there was less of a focus on presentations and more focus on producing and documenting a research roadmap that identifies challenges, different forms of language integration, and relates existing solutions.

The workshop consisted of a keynote by Prof. Dr. Gabor Karsai entitled "Unification or integration: The Challenge of Semantics in Heterogeneous Modeling Language" [5], followed by 2 sessions where the 8 accepted papers were presented. A significant amount of time will be reserved for discussion of paper and their relations to each others. The last session was devoted to a working session dedicated to open discussions of the presented contributions. The discussion was led towards a classification of existing and proposed forms of language integration.

4 Papers Summary

The papers and the associated talks are available in the workshop web site (cf. `http://gemoc.org/gemoc2014/`). Moreover, the proceedings were published by CEUR in the workshop proceedings (cf.`http://ceur-ws.org/Vol-1236/`) which is indexed by DBLP. The reminder of this section presents a brief summary of each paper and the associated discussions.

- [5] ***Keynote* - Unification or integration? The Challenge of Semantics in Heterogeneous Modeling Languages:** The keynote of Prof. Karsai was about the dichotomy between unification versus integration in the context of heterogeneous languages. This talk was specially focused on the language semantics. The problem is initially illustrated by means of three examples where different languages should be integrated or unified. The first example is the problem of modeling and prototyping cyber-physical systems, the second one is the generation of code by using the model-driven development techniques, and the third one is the semantic heterogeneity between different state machines formalisms. Prof. Karsai finishes its presentation by mentioning some of the learned lessons during the construction of these examples that, actually, correspond to real-life projects where he has participated. The main conclusion of the talk is that integration should be problem-specific and the tool support should be carefully considered.

- [7] **Supporting Diverse Notations in MPS' Projectional Editor:** Software languages offer different types of notations to the final users; not only graphical or textual notations –such as UML or SQL respectively–, but also tables and mathematical equations. However, the current languages workbenches usually offer support just for one of those types of notations, and consequently, integration of languages with several notations in the same editor is often pretty hard to achieve. This paper presents the MPS projectional editor for integrating languages despite they offer different types of notations. This presentation was accompanied with a demonstration of several examples of real software languages implemented in this tool.

After the presentation, the discussion of this paper was focused on the meta-language offered by the workbench for specifying the editors. MPS uses a MPS-like meta-language for specifying projectional editors. As a result, languages engineers can customize the meta-language for their particular needs.

- [2] **A Declarative Approach to Heterogeneous Multi-Mode Modelling Languages:** This paper continues the discussion about projectional editors. In this paper, the work is focused on the specification of the interaction between the editor and the abstract syntax tree. This is specially important in this context because a projectional editor is a user interface that directly modifies the corresponding abstract syntax tree based on a semantical definition of certain edition actions. In other words, the user interacts with the AST by means of edition actions defined in the interface and, as a result, there is no parsing like in most of the classical editors.
  In particular, this paper proposes a declarative mechanism for expressing the edition actions. The hypothesis is that term-rewriting can improve the productivity of languages engineers. After the explanation of the approach, the author presented some examples that illustrate the approach.

- [6] **Towards Integrating Modeling and Programming Languages: The Case of UML and Java:** This paper addresses the problem of integrating modeling and programming languages. The main motivation of the work is the existing gap between the modeling activities and the programming activities that leads the fact that certain important decisions such as libraries reuse, are only explicit at the code level. It is important to mention that this approach is specially developed for the the case of integrating UML (fUML in particular) and Java. By using this approach, models written in fUML can invoke methods imported from Java librairies and so directly available in UML (i.e. avoiding black box call) . To do that, the paper presents a process composed of several phases that include some adaptations of the libraries and the definition of the references between the model and the code. The discussion after the presentation of this paper was focused on the possibility of generalizing this approach to other modeling and programming languages. The conclusion is that from the part of the programming language, this approach can be generalized to any language that offers dynamic class loading so libraries can in invoked from fUML. However, from the part of the UML it seems that the approach is quite dependent and there is not a clear idea about how it can be generalized.

- [8] **Understanding Metalanguage Integration by Re-narrating a Technical Space Megamodel:** This paper moved the discussion from the technical part of languages integration, to more pedagogical issues. In particular, the paper begins by explaining the importance of megamodeling in languages integration and then introduces the problem of adopting metamodeling in this community. Basically, the author argues that the complexity encapsulated by megamodels hinders their adoption by final users. To illustrate this issue, the author mentioned several megamodels found in the litera-

ture and shows how difficult is to understand them. The proposal is to use some story-telling approach for communicating this type of complex models. The approach is instantiated in a megamodel developed by the authors for formalizing the relationships among a set of meta-languages.

– [2] **Putting the Pieces Together - Technical, Organizational and Social Aspects of Language Integration for Complex Systems:** This paper reports on an analysis about the different aspects of the integration of software languages. The author discusses about the need of taking into account not only technical aspects, but also social and organizational aspects (internal/external) during the languages integration process. To address this study, this paper presents the results of some interviews applied to different stakeholders in the software industry. The findings can be summarized in three points: (1) at the technical level, pragmatism is the main concern of the stakeholders. It is important to be practical and do not expend unnecessary time; (2) at the organizational level the challenge is more about the construction of the appropriate team that performs the integration; and (3) at the social level, the main objective seems to be the change in the culture of the users so they accept to use the different offered languages.

– [1] **Extensible Global Model Management with Meta-model Subsets and Model Synchronization:** This paper come back to the technical discussion about the globalization of modeling languages. Specifically, the authors use the notion of "global model management" for dealing with the integration (more precisely synchronization) of heterogeneous languages. The hypothesis is that the formalization of the relationships among languages (also called megamodeling in the literature) actually helps for addressing this particular problem. Indeed, based on this assumption the paper presents an infrastructure for the synchronization of languages in the context of the design and implementation of electronic devices. An important characteristic of the approach is that it is based on constraints for restricting the use of the involved languages to enable synchronization.
The discussion after the presentation of the paper was how the presented techniques depend on the application domain. According to the authors, the techniques are completely independent so they can be applied with other case studies.

– [4] **Towards an Ontology-based Approach for Heterogeneous Model Matching:** This paper motivates the integration of heterogeneous languages by means of the analogy of the musicians in an orchestra. In that context, the objective is not to build an instrument that all the musicians can play but to integrate different instruments in such a way that harmonious music can be produced. This can be applied in the context of languages. The objective is not to have generic languages that can be used to describes all the concerns of a system (this is probably impossible), but to have specialized languages for specific domains that can be latter integrated. After that, this paper presents an approach for the integration of languages by using ontologies. The idea is

that an ontology can be used for representing the correspondences between the concepts within different languages. These correspondences are defined at the level of the metamodels and are used at the level of the instances. The discussion in this case was focused on understanding how these techniques depends on the application domain. The conclusions seems to be that these techniques are independent of the application domain.

– [3] **Supporting Debugging in a Heterogeneous, Globally Distributed Environment:** In this paper the authors argue that one of the limitations that hinders the adoption of modeling techniques in the software industry is the lack of tools that fit into the culture of software developers. Particularly, this contribution is focused on debugging since it plays a fundamental role in that culture; the majority of programming languages offer some support for step-wise execution. After this motivation, the paper analyzes the challenges towards the implementation of debugging support in heterogeneous and globally distributed modeling environments and proposes an approach based on query-based debugging. Besides, the notion of *debugging interfaces* is introduced and defined as a set of abstractions for addressing the fact that a modeling environment usually requires support for different formalisms. The discussion afterward was centered in the difference between the challenges of addressing debugging in the development environment versus the production environment.

5 Establishing foundations for engineering the next generation of modeling tools

One of the main objectives of the discussion was to explicitly establish the manner in which the presented approaches contribute to the scope of the workshop, i.e. the globalization of modeling languages. Hence, an important part of the day was dedicated to a discussion that was opened by asking precisely that question to the audience: *How your contribution fits into the vision of globalized modeling languages?*

The first reaction to that question was that the aim of such contributions is to establish the foundations for engineering a **next generation of modeling tools** that are able to offer the functionality of using the different languages that describe the several concerns of a system under construction. However, this claim is not an issue of the integration of several language tools so we can have unified modeling environments supporting many languages. Rather, the need is more about the actual integration of languages in the complete sense of the word so we can have tools that offer unified formalisms for expressing software intensive systems. To do that, software languages should be able to be integrated not only syntactically but also semantically and despite the differences that may exist in terms of implementation platforms and/or paradigms. Rapidly, the discussion was focused on the mechanisms that enable such type of integration. Specifically, the participants of the workshop expressed their perspectives about

the requirements and complexity behind the definition of *languages interfaces*. The main conclusions of these perspectives are:

- **The need of interfaces for static compatibility checking.** Before integrating a set of languages it is desirable to know if they are actually compatible so the integration process can be performed successfully. Consequently, compatibility checking is an important requirement towards the integration of heterogeneous languages. In that context, it is important to take into account that the interfaces should expose the information needed for that verification process.

- **Not all the interfaces are of the same kind.** Compatibility checking is one of the purposes of languages interfaces but it is not the only one. In addition, depending on the purpose of the interface the information that it exposes might be different. As a result, there is a notion of different types of interfaces. During the discussion, two of these different types of interfaces were identified: *tool interfaces* that enable integration at the level of the language tooling, and *user interfaces* that allows the user to interact with the language itself. Besides, one of the participants to the workshop said that, indeed, interfaces can be defined at different levels abstraction. This characterization of the types of interfaces is still an open issue: how to identify and define such taxonomy?

  There is a clear example of interfaces taxonomy in the work of Karsai [5] presented in the keynote. As a result, it makes sense to ask what was the process to identify this specific taxonomy? The answer is that it was based on the observation of the particular situation; the system that is modeled has certain features that are evidently different and that correspond to different concerns of the system. Nonetheless, the types of interfaces are different because they are interfaces for systems, and we are looking for interfaces for languages.

- **Megamodeling may help.** One of the concrete ideas about possible mechanisms for answering these issues is megamodeling since it represents an effort for formalizing the relationships among modeling artifacts. Can the interfaces be understood as special types of relationships among two different languages? Can megamodel help in the management of complex development environments?

- **We need to avoid over-complexing the languages development process.** There is an open question about how the complexity of the development process of languages can be increased with the notion of interfaces. One of the speakers says that the development process is already complex and that we need to be careful because, probably, we can introduce too much additional complexity with this new notion of interfaces.

Acknowledgments

GEMOC 2014 was supported by the GEMOC initiative (df. `http://gemoc.org`) that promotes research seeking 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. Part of this initiative, the workshop was partially supported by the ANR INS project GEMOC (grant #ANR-12-INSE-0011).

GEMOC 2014 was also supported by the ReMoDD project, an NSF funded project (award number CNS-1305381).

References

1. D. Blouin, Y. Eustache, and J.-P. Diguet. Extensible global model management with meta-model subsets and model synchronization. In *Proceedings of the 2nd International Workshop on The Globalization of Modeling Languages*, GEMOC 2014, pages 43–52, Valencia, Spain, September 2014. CEUR.
2. H. Burden. Putting the pieces together - technical, organisational and social aspects of language integration for complex systems. In *Proceedings of the 2nd International Workshop on The Globalization of Modeling Languages*, GEMOC 2014, pages 17–22, Valencia, Spain, September 2014. CEUR.
3. J. Corley and J. Gray. Supporting debugging in a heterogeneous, globally distributed environment. In *Proceedings of the 2nd International Workshop on The Globalization of Modeling Languages*, GEMOC 2014, pages 63–68, Valencia, Spain, September 2014. CEUR.
4. M. E. Hamlaoui, C. Trojahn, S. Ebersold, and B. Coulette. Towards an ontology-based approach for heterogeneous model matching. In *Proceedings of the 2nd International Workshop on The Globalization of Modeling Languages*, GEMOC 2014, pages 53–62, Valencia, Spain, September 2014. CEUR.
5. G. Karsai. Unification or integration? the challenge of semantics in heterogeneous modeling languages. In *Proceedings of the 2nd International Workshop on The Globalization of Modeling Languages*, GEMOC 2014, pages 2–6, Valencia, Spain, September 2014. CEUR.
6. P. Neubauer, T. Mayerhofer, and G. Kappel. Towards integrating modeling and programming languages: The case of uml and java. In *Proceedings of the 2nd International Workshop on The Globalization of Modeling Languages*, GEMOC 2014, pages 23–32, Valencia, Spain, September 2014. CEUR.
7. M. Voelter and S. Lisson. Supporting diverse notations in mps' projectional editor. In *Proceedings of the 2nd International Workshop on The Globalization of Modeling Languages*, GEMOC 2014, pages 7–16, Valencia, Spain, September 2014. CEUR.
8. V. Zaytsev. Understanding metalanguage integration by renarrating a technical space megamodel. In *Proceedings of the 2nd International Workshop on The Globalization of Modeling Languages*, GEMOC 2014, pages 69–77, Valencia, Spain, September 2014. CEUR.