4th International Workshop on Models@run.time

Nelly Bencomo[1] Gordon Blair[1] Robert France[2]
Freddy Muñoz[3], Cédric Jeanneret[4]

[1] Computing Department, Lancaster University, UK
[2] Computer Science Department, Colorado State University, USA
[3] IRISA, INRIA, Equipe Triskell,Rennes, France
[4] Department of Informatics, University of Zurich, Switzerland
{nelly, gordon}comp.lancs.ac.uk, france@cs.colostate.edu,
fmunoz@irisa.fr, jeanneret@ifi.uzh.ch

Abstract. The 4th edition of the workshop Models@run.time was held at the 12th International Conference on Model Driven Engineering Languages and Systems (MODELS). The workshop took place in the city of Denver, Colorado, USA, on the 5th of October 2009. The workshop was organised by Nelly Bencomo, Robert France, Gordon Blair, Freddy Muñoz, and Cédric Jeanneret. It was attended by at least 45 people from more than 10 countries. In this summary we present a synopsis of the presentations and discussions that took place during the 4th International Workshop on Models@run.time.

Keywords: runtime adaptation, MDE, reflection, abstraction.

1 Introduction

The Models@run.time workshop series provides a forum for exchange of ideas on use of models to support software runtime adaptation. The workshops target researchers from different communities, including researchers working on model-driven software engineering, software architectures, computational reflection, adaptive systems, autonomic and self-healing systems, and requirements engineering. This edition of the workshop successfully brought together researchers from different communities: At least forty-five (45) people from nineteen (19) countries attended the workshop. In this workshop we took advantage of the experience gained in previous editions and focused the discussions on the topic: "*Raising the abstraction level*".

In response to the call for papers, sixteen (16) papers were submitted, of which four (4) papers and two (2) tool demonstrations were accepted. Additionally, six (6) papers were invited for poster presentations. Each submitted paper was reviewed by at least 3 program committee members. The papers presented during the workshop are published in a workshop proceedings [1]. Two papers were selected as the best papers. Extended and improved versions of these two papers are published in this post workshop proceedings.

2 Workshop Format and Session Summaries

The workshop activities were structured into presentations, poster, and discussion sessions. In the opening presentation, Robert France set the context of the workshop by summarizing the major results from past workshop editions, presenting the theme of the fourth edition of the workshop "*Raising the abstraction level*", outlining the path to follow during the workshop, and announcing the publication of the special issue on the topic published by *IEEE* Computer in October 2009 [2]. The opening presentation was followed by the papers and posters sessions.

In the paper sessions four (4) papers and two (2) demonstrations were presented. Authors presented their papers in a 20 minutes time slot, which included five minutes for questions and discussion. *Geri George* and *Franck Fleurey* chaired these presentations. In the poster session, six authors presented their work to the workshop attendees.

All presentations were done during the morning to allow enough time for discussion. In the afternoon, the workshop participants formed three groups, where each group was charged with discussing a particularly relevant topic. At the end of the workshop, each group selected a representative who presented the questions raised in the group, and the conclusions reached by the group. More details about the discussion session can be found in section 3. The four (4) paper presentations and the two (2) demos were divided into the following two paper sessions:

Session 1: The use of Computational Reflection

- *Incremental Model Synchronization for Efficient Run-time Monitoring"*, Thomas Vogel, Stefan Neumann, Stephan Hildebrandt, Holger Giese, and Basil Becker.

- *Generating Synchronization Engines between Running Systems and Their Model-Based Views*, Hui Song, Yingfei Xiong, Franck Chauvel, Gang Huang, Zhenjiang Hu, and Hong Mei.

- Demo: *Leveraging Models From Design-time to Runtime. A Live Demo*, Brice Morin, Olivier Barais, Jean-Marc Jézéquel and Grégory Nain.

Session 2: Configuration Management

- *Evolving Models at Run Time to Address Functional and Non-Functional Adaptation Requirements*, Andres J. Ramirez and Betty H.C. Cheng.

- *On the Role of Features in Analyzing the Architecture of Self-Adaptive Software Systems*, Ahmed Elkhodary, Sam Malek and Naeem Esfahani.

- Demo*: Models at Runtime: Service for Device Composition and Adaptation*, Nicolas Ferry, Vincent Hourdin, Stephane Lavirotte, Gaetan Rey, Jean-Yves Tigli, and Michel Riveill.

. The following posters were displayed and presented to the workshop attendees.

- *Using Specification Models for RunTime Adaptations*, Sébastien Saudrais, Athanasios Staikopoulos and Siobhan Clarke.

- *A Model-Driven Configuration Management Systems for Advanced IT Service Management*, Holger Giese, Andreas Seibel and Thomas Vogel.

- *Modeling Context and Dynamic Adaptations with Feature Models*, Mathieu Acher, Philippe Collet, Franck Fleurey, Philippe Lahire, Sabine Moisan and Jean-Paul Rigault.
- *Design for an Adaptive Object-Model Framework: An Overview*, Hugo Ferreira, Filipe Correia and Ademar Aguiar.
- *Management of Runtime Models and Meta-Models in Meta-ORB Reflective Middleware Architecture*, Lucas L. Provensi, Fábio M. Costa and Vagner Sacramento.
- *Statechart Interpretation on Resource Constrained Platforms: a Performance Analysis*, Edzard Hoefig, Peter H. Deussen and Hakan Coskun.

During the afternoon, three discussions groups were established. Each group was charged with discussing a topic based on the questions raised during the presentations and the theme of the workshop – "*Raising the level of abstraction*":

- *The types of models that arise at runtime.*
- *Reasoning and decision making at runtime.*
- *Causal connection between models at runtime and the running system.*

3 Discussions

After the presentations of the morning, the participants were organized into three groups (one group per topic as presented in the previous section). After spending 2 hours discussing the presentations and shared research interests, the groups came back to the meeting room to present a summary of their discussions and conclusions.

Thomas Vogel was the representative for the group that discussed the "*different types of models at runtime"* topic. Thomas presented several categories of models at runtime: (1) abstract and fine-grained models, (2) structural and behavioural models, and (3) dynamic and static models. He emphasized that there are several dependencies among these categories such as critical and non-critical, explicit and implicit, overlapped dependencies (i.e. information replicated among models), and semantic and transitive dependencies. These dependencies raise the following research questions: *How can we identify these dependencies? Are there dependency patterns that appear again and again?* The identification of appropriate dependencies may help determine the impact that a change on one model can have on another. Identifying and managing dependencies will help developers produce the support needed to adapt software based on considerations such as cost and performance trade-off , criticality of functionality, and change rollbacks. Additionally, the support for managing the dependencies and transitive relationships among different categories of models can improve the propagation of correctness, compliance, and consistency criteria from one category to another. Finally, *Thomas* closed the presentation by reminding the audience that the models@run.time community can learn from system management work such as the work carried out by the companies IBM, CA, SAP.

Frank Chauvel was the representative of the group that discussed "*reasoning at runtime*". He started by stating that reasoning is a part of the analysis and planning phases of the control loop typically associated with autonomic systems. Franck then presented four questions that the group identified as main concerns:

(1) *What application domain does the reasoning target?* As reasoning depends on the domain knowledge, it is important to identify in advance the application domain and its constraints. For example, some application domains may require reasoning responses by milliseconds whereas others do it by microseconds. Other constraints may deal with the reasoning's accuracy, the type of reasoning that is required at runtime, or with what can be reasoned beforehand at design time.

(2) *What do we want to achieve with reasoning?* Identifying the goals we seek to reason about and their impact on the actual running system is crucial. It determines the choice of the reasoning strategy to use. For example, reasoning can be performed to maintain over time self-* properties, such as self-adaptation, self-healing, self-configuration,

(3) *How to achieve it?, What techniques are suitable?* According to the goal and the application domain, a variety of techniques may (or not) be suited to perform the needed reasoning. Some techniques are well suited to pre-process and analyze data, to plan and define action sequences, and to deduce and decide (e.g., rule-based systems, goal-based systems, neural networks). When selecting a reasoning technique, it is also useful to ask *What inputs are required by the reasoning engine and how are they provided?* Sometimes domain information may need to be pre-processed and filtered (via model transformation) before reasoning (or before making decision based on that information).

(4) *Which abstraction level should be used?* Recalling the previous presentation, it was shown that different categories of models in different application domains may require different processing and representations. The focus and scope of the reasoning may suggest the abstraction level needed for the different artefacts and decisions. For instance, making decision on varying requirement requires a representation at that abstraction level (functionalities, qualities, etc), whereas making decisions on component swapping and state conservation requires a lower lever representation of components and connectors.

Finally, *Frank* reminded the audience of the need to set clear goals. These goals should ensure the consistency between the answers to the previous questions and clearly states the relation between the application domains, the results of reasoning, and the reasoning techniques that are best suited to accomplish the reasoning task.

Hugo Ferreira was the representative of the group that discussed the causal link between models at runtime and systems. Hugo started by defining a causal connection as a mapping between a model and its base system. The evolution of the model or the system implies that the other should change accordingly. Therefore, causality should be present whenever the system or the model evolves (or many systems and many models). More important, the causal connection should ensure synchronization and consistency at every moment – models and systems are up-to-date and should not cause conflicts. Hugo highlighted some challenges the group identified:

(1) *As the system evolves, how does the runtime model reflect the system changes?* How do we make that possible? Different levels of abstraction help the causal connection to achieve a better granularity level according to the causal goals. For instance, aspects can be helpful for handling crosscutting concerns in causal change (security, authorization, etc). The synchronization rate between models and systems may depend of the goals that causality pursuit. For instance, in some cases an event-

based synchronization may be useful, whereas in other cases, a time frame-based synchronization may be a better choice.

(2) *Who changes the model? Why does it change?* The model can be changed either manually or automatically. These changes may vary on nature according to the goal of the model(s). For instance, some models may change only to reflect the system changes (and perform verification tasks), whereas other may change to adapt the system and change its configuration.

(3) *How do we detect an inconsistency? According to which parameters (efficiency, relevance, degree of inconsistency)?* Inconsistencies may arise when the system and the model change at the same time. It is necessary to define and evaluate means to detect inconsistencies and resolve them.

(4) *In case of inconsistency, which is more valid, the correctness of the model (if any) that controls the system, or the state of the system that should be reflected by the model?* The issue is being able to recognize when to rollback changes in models, and when to rollback changes in the system. There is no real answer to this question, but a case-by-case analysis. Sometimes, we want the system to be "correct" and we may sacrifice the system evolution to maintain correctness. Sometimes, we want the model to be synchronized, and to reflect the system *as is*, then we may sacrifice the model correctness for fast synchronization.

Final Remarks: A general wrap-up discussion was held at the very end of the workshop. The workshop was closed with a warm "thank you" from the organizers to all participants for another successful workshop. After the workshop, the organizers used the feedback from attendees and program committee members to select the best 2 papers. After discussion, the following papers were selected as the best 2 papers:

- *Generating Synchronization Engines between Running Systems and Their Model-Based Views*.
- *Incremental Model Synchronization for Efficient Run-time Monitoring*

Acknowledgments. No workshop is successful by the efforts of only a few people. *We would al*so like to thank the members of the program committee who acted as anonymous reviewers and provided valuable feedback to the authors: *Danilo Ardagna, Fabio M. Costa, Franck Chauvel, Betty Cheng, Peter J. Clark, Frank Fleurey, John Georgas, Jeff Gray, Oystein Haugen, Jozef Hooman, Gang Huang, Paola Inverardi, Jean-Marc Jezequel, Stephane Menoret, Brice Morin, Flavio Oquendo, Daniel Schneider, Rui Silva Moreira, Arnor Solberg, Thais Vasconcelos Batista*. Last but not least, the authors for their interesting submissions are thanked for helping us making this workshop possible.

References

[1] Nelly Bencomo, Gordon S. Blair, Robert B. France, Freddy Muñoz, and Cédric Jeanneret, editors. *Proceedings of the 4th Workshop on Models@run.time, held at the 12th International Conference on Model Driven Engineering Languages and Systems (MODELS'09)*, volume 509 of *CEUR WOrkshop Proceedigns*. CEUR, 2009.

[2] Gordon Blair, Nelly Bencomo, and Robert B. France. Models@ run.time. *Computer*, 42(10):22–27, 2009.