

# Summary of the Workshop Models@run.time at MoDELS 2006

Nelly Bencomo<sup>1</sup>, Gordon Blair<sup>1</sup>, and Robert France,<sup>12</sup>

<sup>1</sup> Lancaster University, Comp. Dep., InfoLab21,  
Lancaster, UK, LA1 4WA,  
Computer Science Department, Colorado State University  
Fort Collins, CO, USA, 80523-1873  
{nelly, gordon}@comp.lancs.ac.uk, france@cs.colostate.edu

**Abstract.** The first edition of the workshop Models@run.time was co-located with the ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems (formerly the UML series of conferences). The workshop took place in the antique city of Genoa, Italy, on the 1st of October, 2006. The workshop was organised by Gordon Blair, Robert France, and Nelly Bencomo. This summary gives an overview an account of the presentations and lively discussions that took place during the workshop.

**Keywords:** model-driven engineering, reflection, run-time systems.

## 1 Introduction

We are witnessing the emergence of new classes of application that are highly complex, inevitably distributed, and operate in heterogeneous and rapidly changing environments. Examples of such applications include those from pervasive and Grid computing domains. These systems are required to be adaptable, flexible, reconfigurable and, increasingly, self-managing. Such characteristics make systems more prone to failure when executing and thus the development and study of appropriate mechanisms for run-time validation and monitoring is needed.

In the model-driven software development area, research effort has focused primarily on using models at design, implementation, and deployment stages of development. This work has been highly productive with several techniques now entering the commercialisation phase. The use of model-driven techniques for validating and monitoring run-time behaviour can also yield significant benefits. A key benefit is that models can be used to provide a richer semantic base for run-time decision-making related to system adaptation and other run-time concerns. For example, one can use models to help determine when a system should move from a consistent architecture to another consistent architecture. Model-based monitoring and management of executing systems can play a significant role as we move towards implementing the key self-\* properties associated with autonomic computing.

## Goal

The goal of this workshop was to look at issues related to developing appropriate model-driven approaches to managing and monitoring the execution and operation of systems. This is the first MoDELS workshop to address this theme. The workshop brought together researchers from a variety of communities including researchers working on model-driven software engineering, software architectures, computational reflection, and autonomic and self healing systems. At least twenty-seven people attended from Austria, Brazil, France, Germany, Italy, Norway, the UK and the US.

The call for papers invited submissions on a number of focus topics including: Relevance and suitability of different model-driven approaches to monitoring and managing systems during run-time, Compatibility (or tension) between different model-driven approaches, Forms of run-time models, Relation with other phases of the software engineering lifecycle, Maintainability and validation of models, and The role of reflection in maintaining the causal connection between models and run-time systems.

In response to the call for papers, nine (9) papers were submitted, of which five (5) papers were accepted for long presentation and two (2) papers for short presentation. Each submitted paper was reviewed by 3 program committee members. After lengthy discussions two papers were chosen as the best papers; the decision took into account the quality of the papers and the relevance of the papers to the goals of the workshop. These papers were extended and improved. The extended versions of these two papers are published in this proceeding.

## 2 Workshop Format

The workshop was designed to facilitate focused discussion on the use of models during run time. It was structured into presentation and work (discussion) sessions. During the morning the guest speaker Prof. Betty Cheng from Michigan State University gave the talk "*Modeling and Analyzing Dynamically Adaptive Software*". This presentation was based on the article "Model-Based Development of Dynamically Adaptive Software" that received an ACM SIGSOFT Distinguished Paper Award in ICSE'06, [1]. Betty presented an approach to creating formal models of adaptive software behaviour. The approach separates the adaptation behaviour and non-adaptive behaviour specifications of adaptive programs, making the models easier to specify and more amenable to automated analysis and visual inspection. Betty presented a process to construct adaptation models, automatically generate adaptive programs from the models, and verify and validate the models. The content of her talk was strongly relevant to the workshop and provided a good kick off and inspiration for lively discussion during the rest of the day.

After Betty's talk, the paper sessions followed. There were two types of presentations, full presentations and short presentations. To ensure effectiveness of the format full presentations were limited to 10 minutes and short presentations were limited to 5 minutes. Both kinds of presentations were followed by 5 minutes of discussion and questions. Furthermore, to facilitate an informed and fruitful discussion, the full presentations were followed by presentations of paper analyses by

assigned independent readers. Each independent reader was someone other than a paper author assigned to discuss the extent to which the paper had addressed the research questions posed in the Call for Papers. After the presentations of the accepted papers, invited speaker Veronique Normand from Thales Research and Technology gave a presentation about the project MODELPLEX.

The afternoon was dedicated to focused discussions on research challenges. Gordon Blair, who was a patient and watchful observer during the morning, took note of the raised questions and comments. Based on his comments and observations, he gave final remarks to shape the discussions of the rest of the afternoon.

The workshop was closed by a general discussion, including an evaluation of the event itself by the participants. Details of the various sessions and other events are provided in Sections 3 and 4 below. The proposed format worked very well, with all attendees contributing to the workshop through full, open, constructive and friendly discussion.

### 3 Session Summaries

Nelly Bencomo welcomed the participants and explained the motivation and format of the workshop.

#### Session 1

The session chair of the session was Robert France who introduced and chaired the discussions of the presentation of the papers:

*"Experiments in Run-Time Model Extraction"*, presented by Jean Bézivin.

*"Applying OMG D&C Specification and ECA Rules for Autonomous Distributed Component-based Systems"*, presented by Jérémy Dubus. Fabio Costa was the second reader.

*"Models at Run-time for sustaining User Interface Plasticity"*, presented by Jean-Sébastien Sottet. Arnor Solberg was the second reader.

#### Session 2

After the coffee break, the second session started. The chair of the session was Nelly Bencomo, who introduced and managed the discussions about the papers:

*"A Run-time Model for Multi-Dimensional Separation of Concerns"*, presented by Ruzanna Chitchyan. Jon Oldevik was the second reader.

*"Towards a More Effective Coupling of Reflection and Run-time Metamodels for Middleware"*, presented by Fabio Costa. Jean-Marc Jezequel was the second reader.

*"Model-driven development of self-managing software"*, presented by Marko Boskovic. Steffen Zschaler was the second reader.

After lunch an invited presentation on the MODELPLEX project was given by Veronique Normand. Several related topics were covered by her presentation. During her talk she discussed how important it is that humans are treated as key parts when making decisions and when defining models. In addition, several perspectives have to be handled, for example design-time system configuration and operation time system

reconfiguration or design-time vs. operation time verification. This last statement was repeated by other presenters during the workshop.

Gordon Blair then provided a summary of the morning. He started off by commenting that we had seen an interesting jigsaw of pieces and it was up to us to put all the pieces together in the afternoon. He followed this by stating that this problem area is probably impossible to solve in the general case and most of the successful work we heard about in the morning narrowed the problem either by focusing on a given application domain and/or by focusing on a particular design methodology (e.g. components, AOSD). He also commented that when addressing the problems it is important to appreciate the reality of distributed systems and solutions must be scalable, must perform well, and must be extensible.

He then highlighted the important role of the software engineering process in identifying complete methodologies for adaptive and autonomous systems (see for example the invited talk by Betty Cheng). It is the premise of the workshop that models have a role throughout such a methodology from early requirements through to run-time.

He commented that many of the contributions in the morning concerned models for run-time, i.e. examples of models that had a role to play during the run-time of the system, whereas what we really need is to step forward and have models at run-time, i.e. models that are an intrinsic part of the systems architecture. This requires a clear understanding of appropriate models, of the running system and of the relationship between them. This leads to the inevitable conclusion that we are concerned with reflection, where the models represent a causally connected self-representation of the system at run-time.

The summary concluded by highlighting some key questions to shape the rest of the discussions:

1. What should a run-time model look like?
2. How can the models be maintained at run-time?
3. What is their role in system validation?
4. What are the best overall model-driven approaches for adaptive and autonomous systems?

In addition, it is important to reflect on the following key meta-level questions:

1. What do we know (useful building blocks)?
2. What do we not know (towards a roadmap)?
3. .... And of course, what should we do next!

## **4 Discussions**

The rest of the afternoon saw the group divided in two discussion subgroups. Both groups shared the same interests and discussed the same set of questions. Summary reports were produced by the leader discussant of each breakout session (Ruzanna Chitchyan and Steffen Zschaler). As the two breakout groups reassembled to summarize their work it was interesting to see how different groups reached very similar conclusions.

When defining what a run-time model looks like both groups coincide in saying that it is related with reflection as it is necessary to have a self representation of the system in operation. A run-time model is no different from any other model where a model is defined as a simpler representation of “reality” that serves a given purpose. The model in this case should be an ongoing representation of the system that is running. There should be a causal connection between the run-time model and the system on execution. The defined model will depend on the problem that is being tackled. Run-time models can offer support to simplify decision making and manipulation, can drive the execution of the application or simply can support for debugging, validation, monitoring, and maintainability.

Each of the questions posed by Gordon cannot be answered without more research. There is need to promote research that explores diverse ways of adapting software during run-time. Furthermore, presentations and discussions make us consider that model-driven approaches offer valuable potential to support run-time adaptability. Model-driven software development would help providing the infrastructure to reconfigure and adapt a run-time system based on input QoS and context based values. The perspective of models at run-time consists in bringing this model-based capability forward to the run-time.

In the end, the workshop itself was evaluated. The organizers asked the participants to provide feedback about the workshop and attendants declared to be very satisfied with the presentations and discussions. It was concluded that the research community should be encouraged to study the issues raised during this workshop.

**Acknowledgments.** We would also like to thank the members of the program committee who acted as anonymous reviewers and provided valuable feedback to the authors: Jan Aagedal, Walter Cazzola, Wolfgang Emmerich, Gang Huang, Jean-Marc Jezequel, Rui Silva Moreira, Marten van Sinderen, Arnor Solberg, and Thaís Vasconcelos Batista. Last but not least, the authors of all submitted papers are thanked for helping us making this workshop possible.

## References

1. J. Zhang and B. H. C. Cheng: Model-Based Development of Dynamically Adaptive Software. in *International Conference on Software Engineering (ICSE'06)*, (China, 2006).