

**Call for Papers 9th workshop on Runtime Verification**  
**RV 2009 Satellite workshop of CAV 2009 June 26 - 28, 2009 Grenoble, France**  
<http://www-rv2009.imag.fr>

### **Aims and Scope**

The objective of RV09 is to bring scientists from both academia and industry together to debate on how to monitor and analyze the execution of programs, for example by checking conformance with a formal specification. The purpose might be testing a piece of software before deployment, detecting errors after deployment in the field and potentially triggering subsequent fault protection actions, or the purpose can be to augment the software with new capabilities in an aspect oriented style. The longer term goal is to investigate whether the use of lightweight formal methods applied during the execution of programs is a viable complement to the current heavyweight methods proving programs correct always before their execution, such as model checking and theorem proving. This year RV 2009 is affiliate to CAV 2009 and is lasting over 3 days with one day for tutorials.

### **Interests**

The subject covers several technical fields as outlined below:

- Specification Languages and Logics. Formal methods scientists have investigated logics and developed technologies that are suitable for model checking and theorem proving, but monitoring can reveal new observation-based foundational logics and problems.
- Aspect-oriented Languages with Trace Predicates. New results in extending aspect languages, such as for example AspectJ, with trace predicates replacing the standard pointcuts. Aspect oriented programming provides specific solutions to program instrumentation and program guidance.
- Program Instrumentation in General. Any techniques for instrumenting programs, at the source code or object code/byte code level, to emit relevant events to an observer.
- Program Guidance in General. Techniques for guiding the behavior of a program once its specification is violated. This includes topics such as fault-protection, self-healing, and diagnosis.
- Combining Static and Dynamic Analysis. Monitoring a program with respect to a specification can have an impact on the monitored program, with respect to execution time as well as memory consumption. Static analysis can be used to minimize the impact by optimizing the program instrumentation. Runtime monitors can be seen as proof obligations left over from proofs - what is left that could not be proved.
- Dynamic Program Analysis. Techniques that gather information during program execution and use it to conclude properties about the program. Algorithms for detecting multi-threading errors, such as deadlocks and data races. Algorithms for generating specifications from runs -- dynamic reverse engineering, this can include program visualization.

### **Invited speakers**

Amir Pnueli (New York University)  
Sriram Rajamani (Microsoft Research India)

### **Paper Submission**

There are two categories of submissions:

1. **Regular Papers:** Submissions, not exceeding fifteen (15) pages using Springer's LNCS format, should contain original research, and sufficient detail to assess the merits and

relevance of the contribution. For papers reporting experimental results, authors are strongly encouraged to make their data available with their submission. Submissions reporting on case studies in an industrial context are strongly invited, and should describe details, weaknesses and strength in sufficient depth. *Simultaneous submission to other conferences with proceedings or submission of material that has already been published elsewhere is not allowed.*

2. **Tool Presentations:** Submissions, not exceeding four (4) pages using Springer's LNCS format, should describe the implemented tool and its novel features. A demonstration is expected at the workshop to accompany a tool presentation. Papers describing tools that have already been presented (in any conference) will be accepted only if significant and clear enhancements to the tool are reported and implemented.

Papers exceeding the stated maximum length run the risk of rejection without review. The review process will include a **feedback/rebuttal** period where authors will have the option to respond to reviewer comments. Papers should be submitted in PDF format. Submission is done with EasyChair. Informations about the submission procedure are available at: <http://wwwrv2009.imag.fr>

### **Important Dates**

Abstract submission: **March 1<sup>st</sup>, 2009**

Paper submission (firm): **March 8, 2009**

Author feedback/rebuttal period: **April 19, 2008**

Notification of acceptance/rejection: **April 26, 2009**

Final version due: **April 26, 2009**

### **Program Chairs**

Saddek Bensalem (Verimag/Université Joseph Fourier, France)  
Doron Peled (University of Bar Ilan, Israel)

### **Program Committee**

Cyrille Valentin Artho (AIST, Japan)  
Howard Barringer (University of Manchester, UK)  
Saddek Bensalem (Verimag/Université Joseph Fourier, France)  
Nikolaj Bjorner (Microsoft Research US)  
Eric Bodden (McGill University, Canada)  
Mads Dam (KTH Stockholm, Sweden)  
Ylies Falcone (Verimag/Université Joseph Fourier, France)  
Bernd Finkbeiner (Saarland University, Germany)  
Cormac Flanagan (university of Santa Cruz, US)  
Pascal Fradet (INRIA, Rhône-Alpes, France)  
Radu Grosu (University of US)  
Klaus Havelund (JPL/NASA, US)  
Moonzoo Kim (KAIST, Korea)  
Insup Lee (University of Pen US)  
Martin Leucker (TUM, Germany)  
Doron Peled (University of Bar Ilan, Israel)  
Mauro Pezzé (University of Milano Bicocca, Italy)  
Shaz Qadeer (Microsoft Research, US)  
Grigore Rosu (University of Illinois Urbana-Champaign, US)  
Henny Sipma (Kestrel Technology, US)  
Scott Smolka (University of Stony Brook, New York, US)  
Oleg Sokolsky (University of Pennsylvania, US)  
Maria Soria (EADS, Germany)  
Scott Stoller (University of Stony Brook, New York, US)