

AperTO - Archivio Istituzionale Open Access dell'Università di Torino

**Preface for the special issue on tool papers of the 23rd International Conference on Coordination Models and Languages, COORDINATION 2021**

**This is a pre print version of the following article:**

*Original Citation:*

*Availability:*

This version is available <http://hdl.handle.net/2318/1889572> since 2023-02-02T12:43:04Z

*Publisher:*

Elsevier

*Published version:*

DOI:10.1016/j.scico.2022.102899

*Terms of use:*

Open Access

Anyone can freely access the full text of works made available as "Open Access". Works made available under a Creative Commons license can be used according to the terms and conditions of said license. Use of all other works requires consent of the right holder (author or publisher) if not exempted from copyright protection by the applicable law.

(Article begins on next page)

# Preface for the Special Issue on Tool Papers of the 23rd International Conference on Coordination Models and Languages, COORDINATION 2021

Giorgio Audrito

*Università di Torino, Italy*

Omar Inverso

*Gran Sasso Science Institute, Italy*

Hugo Torres Vieira

*Evidence Srl, Italy*

---

1 This special issue reports on the tools selected out of the tool track of the  
2 23rd International Conference on Coordination Models and Languages (CO-  
3 ORDINATION 2021), originally programmed in the University of Malta and  
4 held online, on June 14-18, 2021. Since 2019 the conference welcomes sub-  
5 missions describing technological artefacts, including innovative prototypes,  
6 for modelling, analysing, simulating or testing systems in the scope of the  
7 research topics of COORDINATION. The prescribed format of the submis-  
8 sions includes both a short abstract describing the tool and a short video to  
9 showcase the presentation to be carried out at the conference, an innovative  
10 feature that has been highly appreciated by the members of the program  
11 committee.

12 There was a total of 8 submissions to the tool track (about 25% of the to-  
13 tal submissions), and 7 were accepted for publication (about 88% of the total  
14 of tool track submissions and 40% of the number of overall accepted contribu-  
15 tions) as full papers extending the submitted version. For this special issue,  
16 the authors of the more mature contributions were invited to submit their

---

*Email addresses:* [giorgio.audrito@unito.it](mailto:giorgio.audrito@unito.it) (Giorgio Audrito),  
[omar.inverso@gssi.it](mailto:omar.inverso@gssi.it) (Omar Inverso), [hugo.t.vieira@evidence.eu.com](mailto:hugo.t.vieira@evidence.eu.com) (Hugo  
Torres Vieira)

17 tool as an Original Software Publication, a novel format that attends to the  
18 need of disseminating valuable implementation efforts. A total of 6 invita-  
19 tions were issued to COORDINATION 2021 authors, and all 6 contributions  
20 are included in this special issue. Each tool submission was reviewed by  
21 three experts, one of which chosen from the COORDINATION PC members  
22 who reviewed the conference version, and the remaining two among recog-  
23 nised experts in the specific area of the submission, including also reviewers  
24 suggested by the authors themselves.

25 The contributions included in this special issue are the following:

- 26 - Lorenzo Bacchiani, Mario Bravetti, Marco Giunti, João Mota, and  
27 António Ravara. *A Java typestate checker supporting inheritance*. This  
28 contribution presents a tool that allows to perform typestate analysis  
29 over Java programs, recognized by the reviewers as an easy to use tool  
30 and including well documented code. The accompanying paper includes  
31 an illustrative running example that allows the reader to go through  
32 the usage steps while grasping underlying concepts.
- 33 - Davide Basile and Maurice H. ter Beek. *Contract Automata Library*.  
34 Contract automata facilitate the specification, composition, and syn-  
35 thesis of behavioural contracts. This contribution presents Contract  
36 Automata Library (CATLib), a software API supporting the specifi-  
37 cation of contract automata. The accompanying paper discusses the  
38 library architecture, details a usage example and presents recent qual-  
39 itative improvements of the software.
- 40 - Riccardo Bianchini and Francesco Dagnino. *QueryAGT: asynchronous*  
41 *global types in co-logic programming*. QueryAGT aims at high-level  
42 specification of protocols and analysis thereof. It implements in co-  
43 logic programming a novel formalism of global types with asynchronous  
44 communication. The user defines sessions and global types in a simple  
45 language with primitives for type checking. The tool can check different  
46 safety and liveness properties, either in interactive or batch mode.
- 47 - Christian Bartolo Burlò, Adrian Francalanza, Alceste Scalas, Catia  
48 Trubiani and Emilio Tuosto. *PSTMonitor: Monitor Synthesis from*  
49 *Probabilistic Session Types*. This contribution presents PSTMonitor,  
50 a tool for the run-time verification of quantitative specifications of  
51 message-passing applications. Crucially, PSTMonitor uses probabilistic

52 session types to detect executions that deviate from expected proba-  
53 bilistic behaviour. The accompanying paper presents the architecture  
54 and operation of the tool, while assessing the runtime overheads it in-  
55 duces.

56 - Eva Maria Kuehn. *The Peer-Model Tool-Chain*. This paper presents an  
57 open-source implementation of the blackboard-based Peer Model tool  
58 chain. The tool-chain aims at supporting model-driven development  
59 of coordination software and identifying potential issues at an early  
60 stage. A demo video linked from the accompanying paper is available  
61 to further illustrate the proposed methodology.

62 - Rudolf Schlatte, Einar Broch Johnsen, Eduard Kamburjan and S. Lizeth  
63 Tapia Tarifa. *The ABS Simulator Toolchain*. This contribution presents  
64 an executable specification language, for behavioural modelling of dis-  
65 tributed, time- and resource- sensitive communicating systems. The  
66 language provides asynchronous method calls, a resource model, and  
67 support for software product lines. The accompanying paper includes  
68 a review of several case studies available online.

69 We would like to thank all the authors who contributed to this special is-  
70 sue and the members of the Program Committee of COORDINATION 2021  
71 and in particular the chairs, Ferruccio Damiani and Ornela Dardha. We  
72 are also extremely grateful for the work carried out by the referees in the  
73 review of the tools invited for this special issue. Finally, we would like to  
74 acknowledge the support throughout the editorial process from the editors of  
75 Science of Computer Programming Journal, and in particular of the Editors  
76 in chief Mohammad Reza Mousavi and Andrea De Lucia, and also the As-  
77 sociate Editors in chief of the Software Section, Ana Cavalcanti and Marieke  
78 Huisman.