Editorial

Matteo Baldoni*

Dipartimento di Informatica,
Università degli Studi di Torino,
C.so Svizzera 185, I-10149 Torino, Italy
Email: matteo.baldoni@unito.it
*Corresponding author

Luciano Baresi

Dipartimento di Elettronica, Informazione e Bioingegneria,
Politecnico di Milano,
Piazza Leonardo da Vinci 32, I-20133 Milano, Italy
Email: luciano.baresi@polimi.it

Mehdi Dastani

Department of Information and Computing Sciences,
Utrecht University,
Princetonplein 5, 3584 CC Utrecht, The Netherlands
Email: m.m.dastani@uu.nl

Biographical notes: Matteo Baldoni is an Associate Professor at the University of Torino. His research interests include programming and engineering multi-agent systems, socio-technical systems and artificial intelligence. He is in the steering committee of the Italian Association for Artificial Intelligence (AI*IA), of the Engineering Multi-Agent Systems (EMAS) workshop, Normative Multi-Agent Systems (NorMAS) workshop, and Principles and Practice of Multi-Agent Systems (PRIMA) conference. He organised AI*IA 2013, EMAS 2015–2016, NorMAS 2014, DALT 2005–2009, 2012 and EASSS 2009. He was the Program co-Chair of AI*IA 2013 and PRIMA 2016. He has published more than 130 papers and co-edited more than ten books.

Luciano Baresi is a Full Professor at the Politecnico di Milano. He was a Visiting Professor at the University of Oregon, USA and Visiting Researcher at the University of Paderborn, Germany. He was the Program Chair for ICECCS02, FASE06, ICWE07, ICSOC09, SEAMS12 and ESEC/FSE13, and General Chair for WICSA/CompArch16. He is a member of the editorial board of ACM TAAS, IEEE TSC, EAI-endorsed *Transactions on Cloud Systems*, EPiC Series in Computer Science, Springer Requirements Engineering, and *International Journal of Cooperative Information Systems*. He has co-authored more than 130 papers. His research interests are in the broad areas of software engineering, distributed systems, service-based applications and in the different aspects of mobile, self-adaptive, and pervasive software systems.

Mehdi Dastani is an Associate Professor in Computer Science at the Utrecht University. He works in the areas of multi-agent systems, multi-agent logics and multi-agent programming. He has organised many international events

related to multi-agent systems, e.g., various editions of the international conference of Autonomous Agents and Multi-Agent Systems (AAMAS), International Workshop on Multi-Agent Programming (ProMAS), Lorentz seminars on Multi-Agent Organisations and Dagstuhl seminars on Multi-Agent Programming. He was the Program co-Chair of the 6th and 7th International Conference on Fundamentals of Software Engineering, and the Program Co-Chair of the 17th International Conference of Autonomous Agents and Multi-Agent Systems. He has published more than 200 papers and co-edited 14 books.

In 1993, Sommerville emphasised the importance of artificial intelligence methodologies and techniques in the domain of systems engineering albeit the enthusiasm about AI had already faded.[1] Prototypes developed by researchers about knowledge-based systems did not produce commercial products but they produced many good ideas that would have found their way in the subsequent years. Sommerville's (1993) predictions have been confirmed: domain understandings, information collection, system modelling together with a clearer entity and relationship semantics are essential aspects of agile and user-centred methodologies. In the meantime, systems engineering was even more oriented towards distributed and self-* systems. In this perspective, the abstractions for modelling and reasoning about these complex systems become fundamental. Multi-agent systems (MAS) and, in particular, the research area on MAS engineering, can motivate more general techniques and methodologies for systems engineering. Indeed, MAS engineering is a multifaceted, complex task. These systems consist of multiple, autonomous, and heterogeneous agents, and their global behaviour emerges from the cooperation and interaction among the agents. MAS have been widely studied and implemented in academia, but their full adoption in industry is still hampered by the unavailability of comprehensive solutions for conceiving, engineering, and implementing these systems.

Being at the border between software engineering and artificial intelligence, MAS can benefit from both disciplines, but at the same time they lack proper mainstream solutions. For example, even if the artificial intelligence side has been proposing conceptual models for years, there is still a lack of proper abstractions unanimously recognised as effective design solutions for the conception of agents and of their interactions. Similarly, there is still a significant gap between the availability of 'standard' software engineering implementation and validation solutions, and their adoption in the conception of MAS. More recently, the emergence of self-adaptive software systems and, more in general, of the idea of software systems that can change their behaviour at run-time, has imposed MAS as one conceptual solution for their realisation, but it has also emphasised the need for proper and sound engineering solutions. Conversely, design artefacts (e.g., agents or MAS models) can also be used to support and assist the testing and debugging of conventional software, while the use of agent-oriented programming languages results in programs that are more readily verifiable. Thus, many pieces belong to the same puzzle, but significant work is still needed to put them together.

As said, many solutions have already been proposed. They address the use of common software engineering solutions for the conception of MAS, for the use of MAS for improving common software engineering tasks, and also for blending the two disciplines to conceive MAS-centric development processes. Academia has been working on ideas and solutions, which should have exploited by the industry to improve the state of the art. The cross fertilisation is needed to make the two sides of the same coin

cooperate, and a single, common venue can help to exchange ideas, compare solutions, and learn from one another.

The International Workshop on Engineering Multi-Agent Systems (EMAS) aims to be this comprehensive venue, where software engineering and artificial intelligence researchers can meet together and discuss the different viewpoints and findings, and where they can also try to present them to industry. EMAS was created in 2013 as a merger of three separate workshops (with overlapping communities) that focused on the software engineering aspects (AOSE), the programming aspects (ProMAS), and the application of declarative techniques to design, program, and verify (DALT) MAS. The workshop is traditionally co-located with International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS) and this special issue of the *Journal of Agent-Oriented Software Engineering* collects four extended and revised versions of work presented at the EMAS 2015 held in Istanbul.

The paper by Jorge J. Gomez-Sanz, Sandra Garcia-Rodriguez, and Nuria Cuartero-Soler entitled 'Modelling agent oriented solutions for the smart grid', proposes a decentralised agent design for a microgrid that uses agent technology for avoiding dumping the excess of energy to a main powerline. The agent design is made with INGENIAS. The agents connect with a microgrid simulator, the SGSimulator, and issue commands to control different distributed renewable energy sources. This paper illustrates the benefits of coordination of agents in such a scenario through a simple token-based protocol.

The paper by Benjamin Herd, Simon Miles, Peter McBurney, and Michael Luck entitled 'Quantitative analysis of multi-agent systems through statistical verification of simulation traces' focuses on formal verification of large-scale MAS. These systems, due to their complexity, cannot easily be verified by traditional formal verification techniques. Statistical approaches that focus on the verification of individual traces can provide an interesting alternative that circumvents combinatorial explosion. However, due to its focus on finite execution paths, trace-based verification is inherently limited to certain types of correctness properties. This paper shows how, by combining sampling with the idea of trace fragmentation, statistical verification can be used to answer interesting quantitative correctness questions about MAS at different observational levels. The usefulness of the verification approach is illustrated with a simple case study from the area of swarm robotics.

The paper by Brian Logan entitled 'An agent programming manifesto', summarises the invited talk Brian gave at EMAS 2015. Brian focuses on agent programming and gives his point of view about many questions related to the fact that despite the increasing interest in the development of autonomous systems, applications of agent programming are confined to a small number of niche areas, and adoption of agent programming languages in mainstream software development remains limited. This state of affairs is widely acknowledged within the community, and a number of reasons and remedies have been proposed. Brian presents an analysis of why agent programming has failed to make an impact that is rooted in the class of programming problems agent programming sets out to solve, namely the realisation of flexible intelligent behaviour in dynamic and unpredictable environments. Based on this analysis, the author outlines some suggestions for the future direction of agent programming, and some principles that any successful future direction must follow.

Finally, the paper by Nikolaos I. Spanoudakis, Efthymios Floros, Nektarios Mitakidis, and Pavlos Delias entitled 'Validating MAS analysis models with the ASEME methodology', focuses on performance-related non-functional requirements of MAS, an important aspect when designing agent-oriented software. To this end, performance engineering practices provide a useful toolbox, and the simulation of system processes appears eminently suitable. Agent-oriented software engineering methodologies are not directly linked to process simulation features. This paper extends an AOSE methodology for transforming agent role models to process models, and for streamlining the transformation process towards simulation. The method presented in this paper supports process model generation, aiming to support process simulation, and is integrated into a model-driven engineering methodology. The authors use an established process modelling notation (BPMN) as target language for process modelling, and are able to deliver a ready-to-simulate model. Through simulation, an analyst can validate specific system requirements and test scenarios of how the system scales beyond the current requirements. Furthermore, because of their familiarity with process models in the business domain, engineers, managers and stakeholders can seamlessly communicate system designs.

Acknowledgements

We would like to thank all reviewers for their excellent work. Moreover, we would like to thank all the members of the Steering Committee of EMAS for their valuable suggestions and support.

References

Sommerville, I. (1993) 'Artificial intelligence and systems engineering', *Prospects for Artificial Intelligence: Proceedings of AISB '93*, Sloman, A. et al. (Eds.), IOS Press.