Rigorous Engineering of Collective Adaptive Systems Introduction to the 4th Track Edition

Martin Wirsing[1], Rocco De Nicola[2], Stefan Jähnichen[3]

[1] Ludwig-Maximilians-Universität München, wirsing@lmu.de
[2] IMT School for Advanced Studies Lucca, rocco.denicola@imtlucca.it
[3] TU Berlin and FZI Forschungszentrum Informatik Berlin
stefan.jaehnichen@tu-berlin.de

Abstract. A collective adaptive system consists of collaborating entities that are able to adapt in real-time to dynamically changing and open environments and changing needs. Rigorous engineering requires appropriate methods and tools to help ensure that a collective adaptive system lives up to its intended purpose. This note provides an introduction to the 4th edition of the track "Rigorous Engineering of Collective Adaptive Systems" and briefly introduces the panel discussion and its 22 scientific contributions, structured into eight thematic sessions: Design and Validation of Autonomous Systems, Computing with Bio-inspired Communication, New System Models and Tools for Ensembles, Large Ensembles and Collective Dynamics, On the Borderline between Collective Stupidity and Collective Intelligence, Machine Learning for Collective Adaptive Systems, Programming and Analysing Ensembles, and Tools for Formal Analysis and Design.

Keywords: adaptive system, collective system, ensemble, software engineering, formal method, rigorous method, machine learning.

1 Collective Adaptive Systems

Modern IT systems are increasingly distributed and consist of collaborating entities that are able to adapt at runtime to dynamically changing, open-ended environments and to new requirements. Such systems are called Collective Adaptive Systems (CAS) or also ensembles [18, 20]. Examples of CAS are cyber-physical systems, the internet of things, socio-technical systems as well as smart systems and robot swarms.

Rigorous engineering of CAS requires devising appropriate methods and tools to guarantee that such systems behave as expected. To achieve this goal, we need to develop theories for modelling and analysing collective adaptive systems, techniques for programming and running such systems, and specific methods for adaptation, validation and verification while ensuring security, trust and performance.

2 Track Overview

The track "Rigorous Engineering of Collective Adaptive Systems" is a follow-up of four other successful tracks [39], [19], [13], [40] at ISOLA 2014 [26], ISOLA 2016 [27], ISOLA 2018 [28], and ISOLA 2020 [29], respectively. The first track [39] was entitled "Rigorous Engineering of Autonomic Ensembles" and was organised within the activities of the EU-funded research project ASCENS [41]. The latter three tracks [19, 13, 40] addressed the same theme as this year's edition and included research results from several research approaches and projects. Also, a Special Section of the International Journal on Software Tools for Technology Transfer was devoted to the rigorous engineering of collective adaptive systems [12].

The present edition of the track comprises 22 research papers; each of which has undergone a rigorous check by at least two reviewers. During the event, a panel "On the Borderline between Collective Stupidity and Collective Intelligence" took place to discuss the relationships between human and artificial intelligence. The papers were grouped according to seven thematic sessions, viz.: Design and Validation of Autonomous Systems, Computing with Bio-inspired Communication, New System Models and Tools for Ensembles, Large Ensembles and Collective Dynamics, Machine Learning for Collective Adaptive Systems, Programming and Analysing Ensembles, Tools for Formal Analysis and Design.

3 Track Contributions

In this section, the panel discussion and the papers are briefly introduced in the order of their presentations and grouped according to the thematic sessions.

3.1 Design and Validation of Autonomous Systems.

Because of their temporal and spatial dynamism, automotive collective systems are among the most difficult systems to design and validate. The three papers in this session provide novel methods for designing, monitoring, and validating autonomous systems for cars, bikes, and drones.

In their paper "Correct by Design Coordination of Autonomous Driving Systems" [6], Marius Bozga and Joseph Sifakis propose a method for the *correct by design* coordination of autonomous automotive systems. Using assume-guarantee contracts they show that it is practically possible to determine speed control policies for vehicles that are *safe by design.*

Francesca Cairoli, Nicola Paoletti, and Luca Bortolussi do not consider cars but bikes in their paper "Neural Predictive Monitoring for Collective Adaptive Systems" [10]. They present a neural-network learning-based approach, called Neural Predictive Monitoring [5], to preemptively detect violations of requirements for bike-sharing systems, e.g. having bike stations left with no bikes.

Often automotive autonomous systems or more generally distributed cyber-physical systems are virtually synchronous, i.e. they logically behave as if they were synchronous in spite of network delays, and changing execution times. In the paper "An Extension of HybridSynchAADL and Its Application to Collaborating Autonomous UAVs" [22], Jaehun Lee, Kyungmin Bae, and Peter Csaba Ölveczky discuss how to analyze virtually synchronous systems using an extension of the modelling language HybridSynchAADL [21] with compound data types and user-defined functions and illustrate the method by considering a system of collaborating drones for packet delivery.

3.2 Computing with Bio-inspired Communication.

This session focuses on bioinspired computing and presents new approaches for modelling colonies of ants, flocks of birds, and flocks of drones.

The paper "Discrete models of continuous behaviour of collective adaptive systems" [14] by Peter Fettke and Wolfgang Reisig considers artificial ant systems and presents a Petri net approach for modelling the behaviour of the artificial ants, and the causal dependencies between actions, while accounting for continuous movements in discrete models.

In the paper "Modelling Flocks of Birds from the Bottom Up" [11] Rocco De Nicola, Luca Di Stefano, Omar Inverso, and Serenella Valiani propose a novel compositional specification approach for modelling and reasoning about collectives in natural systems. As an example, they incrementally build a bottom-up model of a flock of birds and use a prototype simulator for validating the model in a controlled experiment, where a flock is attacked by a bird of prey and reacts by splitting into smaller groups to reunite when the threat is over.

Andreas Brandstätter and co-authors study flocks of drones and in the paper "Towards Drone Flocking using Relative Distance Measurements" [7] they introduce a method for forming and maintaining a drone flock by considering only relative distance measurements. The proposed approach is fully distributed and can work even in GPS-denied environments.

3.3 New System Models and Tools for Ensembles.

The papers of this session use modal and spatial logic-based methods to specify ensembles of knowledge-based agents, in order to formalise the consciousness of agents and synthesise strategies.

In the paper "Epistemic Ensembles" [15] Rolf Hennicker, Alexander Knapp, and Martin Wirsing study ensembles of knowledge-based agents that, unlike the agents considered in their previous work on ensembles [16, 17], do not use messages to communicate. In this case, information exchange is achieved implicitly through the modification of the knowledge of the agents. Ensemble behaviour is specified in a dynamic logic with compound ensemble actions while specifications are implemented by epistemic processes.

The paper "A modal approach to consciousness of agents" [42] by Chen Yifeng and J. W. Sanders proposes a novel fundamental approach to the notions of awareness and consciousness of agents. Awareness is modelled as a modal operator which satisfies a well-chosen set of basic laws and inequalities. Consciousness is formalised as an iterated form of awareness, more specifically as awareness of awareness.

Maurice ter Beek, Davide Basile, and Vincenzo Ciancia in the paper "An Experimental Toolchain for Strategy Synthesis with Spatial Properties" [34] study the application of strategy synthesis to enforce spatial properties and present the integration of two tools, (i) Contract Automata Library that supports the composition and synthesis of strategies of games modeled in a dialect of finite-state automata, (ii) Voxel-based Logical Analyser, a spatial model checker that supports the verification of properties of (pixels of) digital images. The approach is illustrated through a basic example of the synthesis of strategies on automata that encode the motion of agents in spaces represented by images.

3.4 Large Ensembles and Collective Dynamics.

This section considers the issues connected to the huge number of individuals that a CAS might have.

In the paper "Towards a Kinetic Framework to Model the Collective Dynamics of Large Agent Systems" [30], Stefania Monica, Federico Bergenti, and Franco Zambonelli instantiate the approach based on the kinetic theory of active particles [4] to model and analyse large and decentralized multi-agent systems and use it study cumulative properties of such systems by using statistical techniques that focus on the long-time asymptotic behaviour. As a case study, they show how to derive two asymptotic properties of the symmetric gossip algorithm for multi-agent systems and validate them on a multi-agent implementation of the symmetric gossip algorithm.

Julia Klein and Tatjana Petrov, in the paper "Understanding Social Feedback in Biological Collectives with Smoothed Model Checking" [37], consider biological groups and show that by experimentally observing the collective response of a chosen small set of groups it is possible: (i) to predict the collective response for any given group size and (ii) to infer the desirable group behaviours fitness function which the group robustly performs under different perturbations. They use Smoothed Model Checking, an approach based on Gaussian Process Classification, and specify the fitness function as a template temporal logic formula with unknown parameters. The framework is validated over a case study of a collective stinging defence mechanism in honeybee colonies.

Max Tschaikowski has recently proposed to obtain reliable estimates on global dynamics of agent networks from local agent behavior by replacing dependencies among agents with exogenous parameters, in order to estimate the global dynamics via agent decoupling [36]. The paper "Efficient Estimation of Agent Networks" [23], by Alexander Leguizamon-Robayo and Max Tschaikowski, introduces the notion of estimation equivalence, a model reduction technique for systems of nonlinear differential equations that allows the aforementioned decoupled model to be replaced with a smaller and easier to analyze one. The approach is validated on a multi-class epidemiological SIRS model and is shown to result in a speed-up factor proportional to the number of population classes.

3.5 Panel: On the Borderline between Collective Stupidity and Collective Intelligence.

When observing swarms we might see two different kind of behaviours:

1. the behaviour of individuals appears to be very determined and, above all, the same for all of them. All components follow the same pattern and the behaviour of the swarm as a whole is also very determined. The reaction of the swarm to unknown signals or situations is hardly predictable and rather random and often leads to chaos or even destruction. An example could be the behaviour of lemmings, which for reasons unknown at least to us, join the swarm behaviour and plunge into the sea. We would want to call that *collective stupidity*, although the word stupidity is perhaps not appropriate for a natural behaviour.
2. the behaviour of individual objects is not determined - i.e. each or everyone can do what he or she does "best" - we would call *collective intelligence* if, in the process of achieving a given goal, the feature and the behavior of each individual member of the swarm contribute to the achievement of the goal with its specific characteristics or abilities.

During the panel, Stefan Jähnichen as moderator and the panelists Tomáš Bureš, Thomas Gabor, Joseph Sifakis, Tatjana Petrov, and Franco Zambonelli vividly discussed questions such as "Do we need collective intelligent systems?", "How can we avoid "stupid" swarm behaviour?" or "Can we build systems fostering the collective intelligence of humans?"

3.6 Machine Learning for Collective Adaptive Systems.

This session, consisting of four papers, one of which, for organizational reasons, was presented in the panel session. The paper in the session addresses the issues connected to sub-symbolic artificial intelligence in two complementary ways: using machine learning techniques for supporting collective adaptation and using software development process models for building machine learning systems.

In the paper Ensemble-based modeling abstractions for modern self-optimizing systems" [37], Michal Töpfer and co-authors argue that incorporating machine-learning and optimization heuristics is a key feature of modern smart systems which are to learn over time and optimize their behavior at runtime to deal with uncertainty in their environment. They introduce an extension of their ensemble-based component model DEECo [8] that enables them to use machine-learning and optimization heuristics for managing autonomic component ensembles. An example of how such a model can be beneficially used for modeling access control related problems in the Industry 4.0 settings is provided.

The paper "Attuning Adaptation Rules via a Rule-Specific Neural Network" [9] by Tomáš Bureš and co-authors discusses the use of neural networks in self-adaptive systems. In order to avoid losing some key domain knowledge and improve the learning process, a rule-specific neural network method is introduced that makes it possible to transform the guard of an adaptation rule into a rule for the neural network. The key feature is that rule-specific neural networks are composable and their architecture is driven by the structure of the logical predicates in the adaption rule in question.

To deal with unknowns often online learning is used, but the complexity of online learning increases in the presence of context shifts. In the paper "Measuring Convergence Inertia: Online Learning in Self-Adaptive Systems with Context Shifts" [1], Elvin Alberts and Ilias Gerostathopoulos propose a new metric to assess the robustness of reinforcement learning policies against context shifts and use it to assess the robustness of different policies within a specific class of reinforcement learning policies (multi-armed bandits - MAB) to context shifts. Through an experiment with a self-adaptation exemplar of a web server, they show that their approach is a viable way to inform the selection of online learning policies for self-adaptive systems.

The paper "Capturing Dependencies within Machine Learning via a Formal Process Model" [33] by Fabian Ritz and co-authors defines a comprehensive software development process model for machine learning that encompasses, in a consistent way, most tasks and artifacts described in the literature. In addition to the production of the necessary artifacts, they also consider the generation and validation of fitting descriptions in the form of specifications. They also advocate designing interaction points between standard software development processes and machine learning models throughout their entire life-cycle after initial training and testing.

3.7 Programming and Analysing Ensembles.

In this session, new methods are presented for efficiently running collective adaptive systems and for analysing their quality.

The paper "On Model-based Performance Analysis of Collective Adaptive Systems" [31] by Maurizio Murgia, Riccardo Pinciroli, Catia Trubiani, and Emilio Tuosto is concerned with the analysis of performance properties of CAS. Two recently proposed approaches are considered: one is based on generalised stochastic Petri nets derived from the system specification, while the other is based on queueing networks derived from suitable behavioural abstractions. The relative merits of the two approaches are assessed also by considering a case study based on a scenario involving autonomous robots.

The paper "Programming Multi-Robot Systems with X-KLAIM" [35] by Francesco Tiezzi, Khalid Bourr, Lorenzo Bettini, and Rosario Pugliese also considers software development for robotics applications. It proposes an approach for programming Multi-Robot Systems at a high abstraction level using the programming language X-KLAIM. The computation and communication model of X-KLAIM, based on multiple distributed tuple spaces, allows programs to be coordinated by the same abstractions and mechanisms for both intra- and inter-robot interactions. The feasibility and effectiveness of the proposal are demonstrated in a realistic Multi-Robot Systems scenario.

In the paper "Bringing Aggregate Programming towards the Cloud" [2], Giorgio Audrito, Ferruccio Damiani, Gianluca Torta address the problem of running an Aggregate Programming application on a high-performance, centralized computer such as those available in a cloud environment, in order to manipulate large centralised graph-based data structures across multiple machines, dynamically joining and leaving the computation and have adaptive CAS whose computations dynamically move across the IoT/edge/fog/cloud continuum, according to availability of resources and infrastructures.

3.8 Tools for Formal Analysis and Design.

This section deals with tools and examples of formal design and verification of different kind of collective systems. One considers financial systems, the other deals with cyber-physical systems, while the third one considers agents with opportunistic behaviour.

Bartoletti and his co-authors, in the article "Formal Analysis of Lending Pools in Decentralized Finance" [3] consider decentralized finance applications implemented on blockchain and advocate their formalization and verification. The main contribution is a tool for the formal analysis of lending pools, one of the most popular decentralized finance applications. The tool supports several analyses, including reachability analysis, LTL model checking, and statistical model checking. In the paper, the tool is used to search for threshold and reward parameters that minimize the risk of unrecoverable loans.

In [24] Benjamin Lion, Farhad Arbab, and Carolyn Talcott proposed a compositional approach for modelling distributed cyber-physical systems. There, cyber and physical aspects of a system are described as streams of discrete observations. In the paper in this volume, titled "A Rewriting Framework for Cyber-Physical Systems" [25], the same authors present a rewriting logic implementation of this modelling approach and illustrate it through a case study in which robots move in a common area.

The paper "Model Checking Reconfigurable Interacting Systems" [32] by Nir Piterman, Yehia Abd Alrahman and Shaun Azzopardi deals with reconfigurable multi-agent systems, namely autonomous agents, with integrated interaction capabilities that feature opportunistic interaction. The authors propose a model checker, named R-CHECK, to reason about these systems at both the individual and system levels. The tool supports a high-level input language and allows reasoning about interaction protocols and joint missions, considering reconfiguration, coalition formation and self-organization.

Acknowledgements. As organisers of the track, we would like to thank all authors and panelists for their valuable contributions, all reviewers for their careful evaluations and constructive comments, and all participants of the track for lively discussions. We are also grateful to the ISOLA chairs Tiziana Margaria and Bernhard Steffen for giving us the opportunity to organise this track and to them and Springer–Verlag for providing us with the very helpful Equinocs conference system.

References

1. E. Alberts and I. Gerostathopoulos. Measuring convergence inertia: Online learning in self-adaptive systems with context shifts. In *this volume*, 2022.
2. G. Audrito, F. Damiani, and G. Torta. Bringing aggregate programming towards the cloud. In *this volume*, 2022.
3. M. Bartoletti, J. Chiang, T. Junttila, A. L. Lafuente, M. Mirelli, and A. Vandin. Formal analysis of lending pools in decentralized finance. In *this volume*, 2022.
4. N. Bellomo, D. Burini, G. Dosi, L. Gibelli, D. Knopoff, N. Outada, P. Terna, and M. E. Virgillito. What is life? a perspective of the mathematical kinetic theory of active particles. *Math. Models and Methods in Applied Sciences*, 31(9):1821–1866, 2021.
5. L. Bortolussi, F. Cairoli, and N. Paoletti. Neural predictive monitoring and a comparison of frequentist and Bayesian approaches. *Int. J. Softw. Tools Technol. Transf.*, 23(4):615–640, 2021.
6. M. Bozga and J. Sifakis. Correct by design coordination of autonomous driving systems. In *this volume*, 2022.
7. A. Brandstätter, S. A. Smolka, S. D. Stoller, A. Tiwari, and R. Grosu. Towards drone flocking using relative distance measurements. In *this volume*, 2022.
8. T. Bures, I. Gerostathopoulos, P. Hnětynka, J. Keznikl, M. Kit, and F. Plášil. DEECO: an ensemble-based component system. In P. Kruchten, D. Giannakopoulou, and M. Tivoli, editors, *CBSE'13, Proceedings of the 16th ACM SIGSOFT Symposium on Component Based Software Engineering, part of Comparch '13, Vancouver, BC, Canada, June 17-21, 2013*, pages 81–90. ACM, 2013.
9. T. Bureš, P. Hnětynka, M. Kruliš, F. Plášil, D. Khalyeyev, S. Hahner, S. Seifermann, M. Walter, and R. Heinrich. Attuning adaptation rules via a rule-specific neural network. In *this volume*, 2022.
10. F. Cairoli, N. Paoletti, and L. Bortolussi. Neural predictive monitoring for collective adaptive systems. In *this volume*, 2022.
11. R. De Nicola, L. Di Stefano, O. Inverso, and S. Valiani. Modelling flocks of birds from the bottom u. In *this volume*, 2022.
12. R. De Nicola, S. Jähnichen, and M. Wirsing. Rigorous engineering of collective adaptive systems: special section. *Int. J. Softw. Tools Technol. Transf.*, 22(4):389–397, 2020.
13. R. De Nicola, S. Jähnichen, and M. Wirsing. Rigorous engineering of collective adaptive systems – introduction to the 2$^{\text{nd}}$ track edition. *In [28]*, pages 3–12, 2018.
14. P. Fettke and W. Reisig. Discrete models of continuous behavior of collective adaptive systems. In *this volume*, 2022.
15. R. Hennicker, A. Knapp, and M. Wirsing. A modal approach to consciousness of agents. In *this volume*, 2022.
16. R. Hennicker and M. Wirsing. Dynamic logic for ensembles. In *[28]*, pages 32–47, 2018.
17. R. Hennicker and M. Wirsing. A dynamic logic for systems with predicate-based communication. In *[29]*, pages 224–242, 2020.

18. M. M. Hölzl, A. Rauschmayer, and M. Wirsing. Engineering of software-intensive systems: state of the art and research challenges. In *[38]*, pages 1–44. Springer, 2008.
19. S. Jähnichen and M. Wirsing. Rigorous engineering of collective adaptive systems – track introduction. *In [27]*, pages 535–538, 2016.
20. S. Kernbach, T. Schmickl, and J. Timmis. Collective adaptive systems: challenges beyond evolvability. *CoRR abs/1108.5643*, 2011.
21. J. Lee, S. Kim, K. Bae, and P. C. Ölveczky. HybridSynchAADL: Modeling and formal analysis of virtually synchronous CPSs in AADL. In A. Silva and K. R. M. Leino, editors, *Proc. Computer Aided Verification - 33rd International Conference, CAV 2021, Part I*, volume 12759 of *Lect. Notes Comp. Sci.*, pages 491–504. Springer, 2021.
22. J. Lee, S. Kim, K. Bae, and P. C. Ölveczky. An extension of HybridSynchAADL and its application to collaborating autonomous UAVs. In *this volume*, 2022.
23. A. Leguizamon-Robayo and M. Tschaikowski. Efficient estimation of agent networks. In *this volume*, 2022.
24. B. Lion, F. Arbab, and C. L. Talcott. A semantic model for interacting cyber-physical systems. In J. Lange, A. Mavridou, L. Safina, and A. Scalas, editors, *Proc. 14th Interaction and Concurrency Experience, ICE 2021*, volume 347 of *EPTCS*, pages 77–95, 2021.
25. B. Lion, F. Arbab, and C. L. Talcott. A rewriting framework for cyber-physical systems. In *this volume*, 2022.
26. T. Margaria and B. Steffen, editors. *Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change - 6th International Symposium, ISoLA 2014, Part I*, volume 8802 of *Lecture Notes in Computer Science*. Springer, 2014.
27. T. Margaria and B. Steffen, editors. *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 7th International Symposium, ISoLA 2016, Part I*, volume 9952 of *Lecture Notes in Computer Science*. Springer, 2016.
28. T. Margaria and B. Steffen, editors. *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 9th International Symposium, ISoLA 2018, Part III*, volume 11246 of *Lecture Notes in Computer Science*. Springer, 2018.
29. T. Margaria and B. Steffen, editors. *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 10th International Symposium, ISoLA 2020, Part II*, volume 12477 of *Lecture Notes in Computer Science*. Springer, 2020.
30. S. Monica, F. Bergenti, and F. Zambonelli. Towards a kinetic framework to model the collective dynamics of large agent systems. In *this volume*, 2022.
31. M. Murgia, R. Pinciroli, C. Trubiani, and E. Tuosto. On model-based performance analysis of collective adaptive systems. In *this volume*, 2022.
32. N. Piterman, Y. Abd Alrahman, and S. Azzopardi. Model checking reconfigurable interacting systems. In *this volume*, 2022.
33. F. Ritz, T. Phan, A. Sedlmeier, P. Altmann, J. Wieghardt, R. Schmid, H. Sauer, C. Klein, C. Linnhoff-Popien, and T. Gabor. Capturing dependencies within machine learning via a formal process model. In *this volume*, 2022.
34. M. ter Beek, D. Basile, and V. Ciancia. An experimental toolchain for strategy synthesis with spatial properties. In *this volume*, 2022.
35. F. Tiezzi, K. Bourr, L. Bettini, and R. Pugliese. Programming multi-robot systems with X-KLAIM. In *this volume*, 2022.

36. M. Tschaikowski. Over-approximation of fluid models. *IEEE Trans. Autom. Control.*, 65(3):999–1013, 2020.
37. M. Töpfer, M. Abdullah, T. Bureš, P. Hnětynka, and M. Kruliš. Ensemble-based modeling abstractions for modern self-optimizing systems. In *this volume*, 2022.
38. M. Wirsing, J.-P. Banâtre, M. M. Hölzl, and A. Rauschmayer, editors. *Software-Intensive Systems and New Computing Paradigms - Challenges and Visions*, volume 5380 of *Lecture Notes in Computer Science*. Springer, 2008.
39. M. Wirsing, R. De Nicola, and M. M. Hölzl. Rigorous engineering of autonomic ensembles – track introduction. *In [26]*, pages 96–98, 2014.
40. M. Wirsing, R. De Nicola, and S. Jähnichen. Rigorous engineering of collective adaptive systems – introduction to the 3rd track edition. *In [29]*, pages 161–170, 2020.
41. M. Wirsing, M. M. Hölzl, N. Koch, and P. Mayer, editors. *Software Engineering for Collective Autonomic Systems – The ASCENS Approach*, volume 8998 of *Lecture Notes in Computer Science*. Springer, 2015.
42. C. Yifeng and J. W. Sanders. Epistemic ensembles. In *this volume*, 2022.