A Quantitative Approach to the Design and Analysis of Collective Adaptive Systems

Luca Bortolussi, Rocco De Nicola, Nicolas Gast, Stephen Gilmore, Jane Hillston, Mieke Massink, Mirco Tribastone

***Abstract*—Reasoning about quantitative properties of collective adaptive systems is made difficult by the typically massive size of the models used for the analysis. In this paper, we present our envisaged approach to the prediction and control of the behaviour of large-scale systems, based on formal methods for their rigorous and unambiguous specification, and an array of scalable and accurate underlying analytical tools.**

Collective adaptive systems (CAS) consist of a large number of spatially distributed heterogeneous entities with decentralised control and varying degrees of complex autonomous behaviour. It is of fundamental importance that a thorough *a priori* analysis of their design is carried out to investigate all aspects of their behaviour, including quantitative and emergent aspects, before they are put into operation. This is to avoid costly re-designs or re-deployments that are necessitated if an envisaged solution does not turn out to fulfil the desired quality of service. Therefore, performing analysis as early as possible in the development lifecycle gives designers high confidence that, once operational, CAS can adapt to changing requirements autonomously without operational disruption. Unfortunately, the defining characteristics of these systems mean that their behaviour is often highly unpredictable or counter-intuitive. Formal, scalable, quantitative analysis, which provides multiple perspectives on system behaviour while being based on well-established reasoning techniques, is therefore imperative to master such complex systems.

Our main objective is the development of an innovative design framework that provides a formal specification language for CAS and a large variety of tool-supported, scalable quantitative analysis and verification techniques. Specifically, our framework will consist of the following main elements.

CAS-SCEL: CAS-SCEL is a specification language based on process algebra (e.g., [1]). This is motivated by the possibility of performing *compositional design*. For instance, a model of a CAS with n entities, each associated with a *local* specification denoted by $E_1, \ldots, E_n$, would be described by an equation in the form $E_1 \parallel \cdots \parallel E_n$. Here the *parallel operator* "$\parallel$" informally means that the local behaviour of each entity can be influenced by the fact that it is operating in a context where other entities are present, either with shared or with conflicting goals and behaviours. Since process algebra can be seen as kernels of more expressive higher-level programming languages, such compositional specification may naturally lead to a component-based implementation of the software that will ultimately realise the design. In addition, process algebra offer *compositional reasoning*. As an example, a property that is found to hold for a specific entity (or group of entities) can carry over in every context in which the entity is operating.

This feature is at the basis of a recent technique that aims at a scalable analysis of large-scale process algebra models. In a CAS, it is not uncommon to find large groups of entities that are identical, or that can be reasonably modelled as being so. (The CAS may still be fundamentally heterogeneous due to the presence of distinct groups of such entities.) For instance, a group of n entities, each modelled by some specification E of size k, could be written

$$\underbrace{E \parallel \cdots \parallel E}_{n \text{ times}}.$$

Such a model would be ordinarily associated with an underlying mathematical object (e.g., a continuous-time Markov chain) whose size grows at worst exponentially as k^n. Instead, a *fluid* interpretation allows for an efficient approximation, based on a system of ordinary differential equations. This is fundamentally independent from n, and only grows linearly with k, i.e., the size of the single agent (e.g., [2]). By virtue of compositional reasoning, regardless of the context in which this group operates, such an approximation can always be applied.

CAS-SCEL will also feature a fluid interpretation which allows the modeller to infer, in a scalable manner, the global, macroscopic behaviour of a group of entities from the local, microscopic description of the constituting components.

Scalable representation of space: CAS are inherently spatially distributed. Although in some cases it is possible to abstract away from space-related properties (for instance, when the system can be modelled as being well-mixed), in general locality and mobility patterns play an important role in determining the overall system behaviour. Within CAS-SCEL we will allow the description of systems with explicit spatial aspects. An expected novel contribution in this respect is a general methodology for spatial modelling based on sound mathematical foundations. This will be achieved by identifying analytical approaches to spatial representation that are most amenable to use in the context of modelling CAS, where efficient evaluation is a prime consideration, and embed them in CAS-SCEL using appropriate formal constructs which guarantee soundness of the approximation.

Several representations of space should be explored as we deem it unlikely that one form of representation and analysis

Luca Bortolussi is with the Department of Mathematics and Geosciences, University of Trieste, Italy, and with CNR-ISTI, Pisa, Italy.

Rocco De Nicola is with IMT Lucca, Italy.

Nicola Gast is with EPFL, Switzerland.

Stephen Gilmore and Jane Hillston are with the School of Informatics, University of Edinburgh, Scotland.

Mieke Massink is with CNR-ISTI, Pisa, Italy.

Mirco Tribastone is with the Department for Informatics, Ludwig Maximilians University of Munich, Germany.

will be able to capture all situations presented by CAS. Thus, space will be modelled in different ways, including regular grids, continuous representations, and patch-based models. Associated with them are different quantitative approaches to model dynamics in space: random walks and cellular automata on grids, partial differential diffusion equations or Brownian motion on continuous space, differential equations or Markov processes on patches (e.g., [3]).

Scalable verification combining micro and macro perspectives: We will develop novel techniques for the verification of large-scale systems described with CAS-SCEL, building on recent work that verifies stochastic temporal logic formulae using *fluid model-checking*, a model-checking approach that exploits fluid approximation [4]. It allows an abstraction of the global system behaviour in order to study the evolution of a single agent by decoupling its behaviour from the evolution of its environment, dramatically reducing the cardinality of the state space by several orders of magnitude. Such single-agent models are time-inhomogeneous CTMCs for which no known algorithms are available as yet. The development of such model-checking algorithms will enable the formulation and verification of properties of single agents that are dependent on the global behaviour of the system, thus bridging individual (micro) level and population (macro) level views of CAS in a novel way. Furthermore, we will extend scalable model-checking techniques with appropriate spatial logics to combined spatial and timed properties of CAS. Inspired by the available literature on spatial extensions of stochastic temporal logics for classical model-checking [5], we will investigate and develop variants that match the spatial primitives that will be added to CAS-SCEL. Our main goal is to extend fluid model checking techniques such that space-related properties can be verified in a scalable manner.

Design workflow and analysis pathway: To improve the usability of our results by interested CAS stakeholders, our techniques will be implemented as software tools, building on already existing frameworks [6]. Furthermore, we will specify a design workflow which details the steps which need to be taken to create collaborative systems which can be operated safely and cost-effectively. The design workflow will list types of analysis which should be applied to the design of a collaborative system such as: static analysis, reachability analysis, deadlock detection, a single simulation study, simulation trace analysis, simulation ensemble studies, variability analysis, scalability analysis, large-scale mean field and fluid flow analysis, parameter estimation, qualitative model checking, quantitative model checking, fluid model checking, and other analyses. Not all of these types of analysis are applicable to every model. The analysis pathway will guide users through this process, directing them first to the simple and inexpensive analysis which must be done on models, then leading them to the more computational expensive analysis which need only be done if initial inexpensive checks have passed. The analysis pathway will also clarify which kinds of analysis are appropriate for which models and maintain a record of the user's progress through the analysis process.

Case studies: We will consider case studies within the domain of *smart cities*, focussing on smart grids, where the emphasis is on the optimal control of energy production, consumption, and distribution; and on smart public transportation, which aims at efficient planning and management by means of integrated computer infrastructures.

Smart grids: Electricity networks exhibit multiple organisational scales, for example: local or large-scale production and decentralised consumption. Since the instantaneous supply of electricity must always meet the constantly changing demand, operation of an electric power system involves a complex process of forecasting production and demand. A way for a distribution system operator to match the consumption with the production is to introduce active demand/response by sending prices or interruption signals to users or by using large-scale storage systems. Consumers also have their own motivations and distinct reactions to signals, such as using battery or reducing consumption. Building efficient mechanisms and distributed control strategies that lead to efficient emergent behaviour still remains an open issue.

Smart public transport: About 500 large cities worldwide are adopting fully automated bike-sharing systems, as a means to reducing vehicular traffic, pollution, and energy consumption. Research challenges are the construction of efficient redistribution policies and the incentive mechanisms to improve load balancing. These concerns also apply to other forms of public transportation, in particular, buses. In both cases, customer satisfaction plays an important role, as providers want to avoid disappointing experiences such as a user wanting to hire a bike finding a station empty, or finding it full when she wants to return the bike. Similarly, bus providers aim at fully utilising the capacity of the vehicles whilst ensuring satisfactory frequencies of service, tailored to the possibly variable passengers' needs.

In our vision, these case studies, and many more, can be analysed within our framework for the rigorous and effective prediction of their quantitative behaviour, leading to robust and cost-effective designs and implementations, for the benefit of CAS providers and consumers alike.

Acknowledgement

This work is partially supported by the EU project QUANTICOL, 600708.

References

[1] R. Milner, *Communication and Concurrency*. Prentice-Hall, 1989.

[2] J. Hillston, "Fluid flow approximation of PEPA models," in *Proceedings of the Second International Conference on the Quantitative Evaluation of Systems*. Torino, Italy: IEEE Computer Society Press, Sep. 2005, pp. 33–43.

[3] A. Chaintreau, J.-Y. Le Boudec, and N. Ristanovic, "The age of gossip: spatial mean field regime," in *SIGMETRICS/Performance*, 2009, pp. 109–120.

[4] L. Bortolussi and J. Hillston, "Fluid model checking," in *CONCUR*, 2012, pp. 333–347.

[5] R. De Nicola, J.-P. Katoen, D. Latella, M. Loreti, and M. Massink, "Model checking mobile stochastic logic," *Theor. Comput. Sci.*, vol. 382, no. 1, pp. 42–70, Aug. 2007.

[6] M. Tribastone, A. Duguid, and S. Gilmore, "The PEPA Eclipse Plug-in," *Performance Evaluation Review*, vol. 36, no. 4, pp. 28–33, March 2009.