# Challenges for Quantitative Analysis of Collective Adaptive Systems

**Citation for published version:**
Hillston, J 2014, Challenges for Quantitative Analysis of Collective Adaptive Systems. in M Abadi & A Lluch Lafuente (eds), *Trustworthy Global Computing: 8th International Symposium, TGC 2013, Buenos Aires, Argentina, August 30-31, 2013, Revised Selected Papers.* Lecture Notes in Computer Science, Springer International Publishing, pp. 14-21. https://doi.org/10.1007/978-3-319-05119-2_2

**Digital Object Identifier (DOI):**
10.1007/978-3-319-05119-2_2

**Link:**
Link to publication record in Edinburgh Research Explorer

**Document Version:**
Early version, also known as pre-print

**Published In:**
Trustworthy Global Computing

# Challenges for Quantitative Analysis of Collective Adaptive Systems

Jane Hillston

LFCS, School of Informatics, University of Edinburgh, Scotland, UK
jane.hillston@ed.ac.uk

## 1 Introduction

We are surrounded by both natural and engineered *collective systems*. Such systems include many entities, which interact locally and, without necessarily having any global knowledge, nevertheless work together to create a system with discernible characteristics at the global level; a phenomenon sometimes termed *emergence*. Examples include swarms of bees, flocks of birds, spread of disease through a population, traffic jams and robot swarms. Many of these systems are also *adaptive* in the sense that the constituent entities can respond to their perception of the current state of the system at large, changing their behaviour accordingly. Since the behaviour of the system is comprised of its constituent entities this brings about a change in the system, thus creating a feedback loop. For example, when a disease is spreading epidemically people adjust their behaviour to reduce contact with others; consequently the spread of the disease may diminish.

Increasingly IT systems are being build from large numbers of autonomous or semi-autonomous components which, together with a large population of users, makes a collective system. For example, in Edinburgh bus are equipped with GPS sensors, and bus stops have display boards, which inform users of the likely arrival time of the next bus on various routes. Bus users can choose which route to take for their journey based on the given information. As in this example, collective IT systems are often embedded in our environment and need to operate without centralised control or direction. Moreover when conditions within the system change it may not be feasible to have human intervention to adjust behaviour appropriately. For example, it would be desirable for a major traffic incident that re-routes some buses to be indicated on the information boards. For this to happen in general systems must be able to adapt autonomously.

What we are starting to witness is the establishment of what Robin Milner called the *informatics environment*, in which pervasive computing elements are embedded in the human environment, invisibly providing services and responding to requirements [20]. Such systems are now becoming the reality, and many form collective adaptive systems, in which large numbers of computing elements collaborate to meet the human need. The smart bus system described above is one example, and there are many others in the realm of "Smart Cities" where information flows to and from users to enhance access and efficient use of resources.

Performance modelling aims to construct models of the dynamic behaviour of systems in order to support the fair and timely sharing of resources. Performance problems typically arise when there is contention for resources and this can impede the smooth running of a system and lead to user dissatisfaction. In the informatic environment, where the system itself is often almost invisible to the user, it is essential that the possible behaviour is thoroughly explored before systems are deployed. Performance analysis appears in many guises and may more generally be termed *quantitative analysis*, as it encompasses many quantified questions about the dynamic behaviour of systems. For example:

**Capacity Planning:** how many clients can the existing server support and still maintain reasonable response times? or how many buses do I need in order to maintain service at peak time in a smart urban transport system.

**System Configuration:** in a mobile phone network how many frequencies do I need in order to keep the blocking probability for new calls low? or what capacity do I need at the stations in a bike sharing scheme in order to minimise the extent to which bikes have to be relocated by truck to meet user demand?

**System Tuning:** in a flexible manufacturing system, what speed of conveyor belt will minimise robot idle time and maximum throughput whilst avoiding damaged goods? or what strategy can I use to maintain supply-demand balance within a smart electricity grid?

Markovian-based discrete event models have been applied to the performance prediction of computer systems since the mid-1960s and communication systems since the early 20th century. Originally queueing networks were primarily used to construct models, and sophisticated analysis techniques were developed. This approach is challenged by features of modern distributed systems, and there has been a shift towards the use of formal methods, in which formal language are enhanced with quantitative information such as durations and probabilities. Examples include Generalised Stochastic Petri Nets [1], and Stochastic Process Algebras such as EMPA [2], IMC [11] and PEPA [12]. From these high-level system descriptions the underlying mathematical model (Continuous Time Markov Chain (CTMC)) can be automatically generated via the formal semantics.

## 2 Progress in recent years

A key feature of collective systems is the existence of populations of entities who share certain characteristics. Attempts to model such systems without high-level modelling support are likely to be time-consuming and error-prone. In contrast, high-level modelling formalisms allow this repetition to be captured at the high-level rather than explicitly, and often support hierarchical and compositional development of models.

In particular process algebras are well-suited for constructing models of collective adaptive systems (CAS):

- These formal languages were originally developed to represent concurrent behaviour compositionally and CAS are highly concurrent systems.
- The compositional structure of the process algebra allows the interactions between individuals to be captured explicitly. In the context of CAS individuals of the same type may be regarded as a subpopulation with limited interaction between entities but all sharing the same pattern of interaction with other populations.
- Stochastic process algebras (SPAs) provide extensions of classical process algebras that allow the dynamics of system behaviour to be captured; moreover there are established mechanisms to automatically generate an underlying mathematical model from the process algebra description.
- In SPAs such as PEPA, state-dependent functional rates mean that the rate or probability with which an event occurs may depend on the current state of the system and this can allow adaptation to be captured [14].
- The languages are equipped with formal apparatus for reasoning about the behaviour of systems, including equivalence relations, formally defined abstraction mechanisms and mappings to model checkers such as PRISM [16].

As originally defined, an SPA model is equipped with a structured operational semantics which facilitates the automatic generation of a CTMC. In this case the global state of the system is the composition of the local states of all the participating components. When the size of the state space is not too large the CTMC is represented explicitly as an infinitesimal generator matrix, which is an $N \times N$ matrix, where $N$ is the number of distinct states. Based on this matrix and linear algebra the CTMC can be subjected to a numerical solution which determines a steady state or transient probability distribution over all possible states. From this, performance indices such as throughput, utilisation and response time can be derived.

Alternatively the CTMC may be studied using stochastic simulation. This avoids the explicit construction of the entire state space, as states are generated on-the-fly as the simulation runs. Each run generates a single trajectory through the state space. Now, performance indices are derived from measurement of the behaviour of the simulation model and many runs are needed in order to obtain statistically meaningful estimates of performance measures.

Like all discrete state representations, performance modelling formalisms and CTMCs suffer from the problem of *state space explosion*: the mathematical structured required to analyse the system become so large that it is infeasible to carry out the analysis. As the size of the state space becomes large it becomes infeasible to carry out numerical solution of the CTMC and extremely time-consuming to conduct stochastic simulation. This poses a severe challenge for the analysis of collective systems, which by their nature typically contain very large numbers of entities.

The discrete state interpretation of SPA models is focussed on treating the instances of components as individuals. An alternative, more compact representation can be obtained if we move away from capturing each individual but instead work at the level of the subpopulations. This is clearly an abstraction,

and some information is lost, but it has the advantage that substantially larger systems can be considered.

The first step of our approach to analysing collective behaviour is to make a *counting abstraction* and view the system not in terms of the individual components but in terms of proportions within the subpopulations [15]. This is shown schematically in Figure 1.

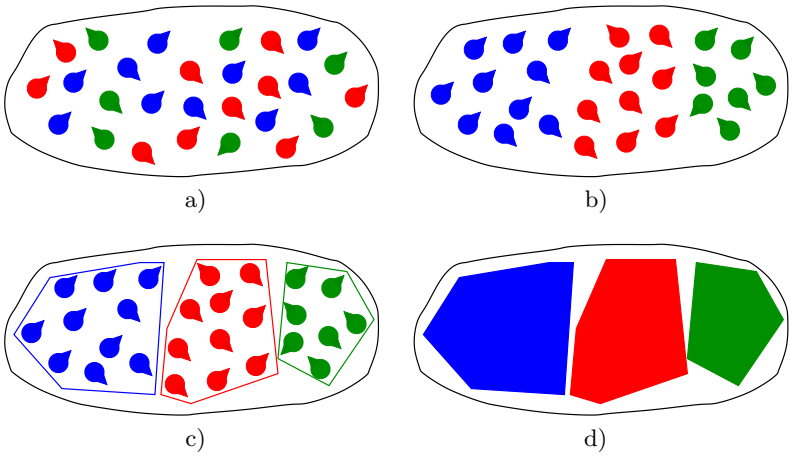


**Fig. 1.** Schematic representation showing the counting abstraction: a)–b)subpopulations are identified within the CAS; c)–d) rather than explicit counts, these are represented as proportions of the population as a whole.

Initially this produces a *state aggregation*: a more compact discrete representation of the system. A further shift in perspective leads us to consider the evolution of the system as *continuous* rather than discrete. In this case the events in the system are aggegrated, and captured by ordinary differential equations which represent the average behaviour of the system, in terms of the proportions of components which exhibit each possible local behaviour or state and how these proportions vary over time [13]. This is termed a *fluid* or *mean field approximation* [4].

Just as the discrete representation of the CTMC can be automatically generated from the structured operational semantics of PEPA models [12], the ODEs which give the fluid approximation of a PEPA model can similarly be derived from structured operational semantics [22]. Moreover the derived vector field $\mathcal{F}(x)$, gives an approximation of the expected count for each population over time and *fluid rewards*, from which performance indices can be derived, can be safely calculated from the fluid expectation trajectories [21]. Furthermore, vector fields have been defined to approximate higher moments [9], such as variance and skew, allowing more accurate estimates of the performance of a system

to be derived and more sophisticated measures, such as passage times, can be approximated in an analogous way [10].

This approach is ideally suited to the analysis of collective systems, which would typically overwhelm existing techniques — the necessary state space could not even be expressed, never mind analyse. Examples of systems which have been studied using this approach include an emergency egress system [18], smart buildings [19], data flows in wireless sensor networks [8] swarm robots [17], and internet worm attacks [6].

## 3 Remaining challenges

The fluid approximation approach coupled with formal model description in terms of a stochastic process algebra has opened new opportunities for quantified formal analysis of collective systems. This work provides a basic framework and firm foundation for the modelling of systems with collective behaviour. Nevertheless, there remain a number of challenges, especially when we consider systems which also consider adaptive behaviour. In particular, based on our experiences of modelling smart city applications within the QUANTICOL project[1] we would highlight:

- Spatial aspects;
- Richer forms of interaction and adaptation; and
- Extending model checking capabilities.

### 3.1 Modelling space

Whilst fluid approximation of SPA models has been successfully used to model collective systems, it should be recognised that there is an implicit assumption within the approach that all components are co-located. This means that all components have the opportunity to interact if their specified behaviour allows it.

However, many collective systems, particularly in the context of smart cities, have behaviour which is partially governed by the spatial arrangement of the components. Interactions may only be allowed for entities which are within a certain physical distance of each other, or space may be segmented in such a way that even physically close entities are unable to communicate. Furthermore movement can be a crucial aspect of the behaviour of entities within the system. Capturing and analysing systems with characteristics like these require that space must be included explicitly within the modelling formalism, and the same component in different locations will be distinguished. This poses significant challenges both of model expression and model solution. There is a danger that as we distinguish subpopulations by their location, we no longer have a large enough population to justify the fluid approximation.

Initial work is exploring the use of time scale decompositions, partial differential equations and diffusion models but much more work is needed.

[1] www.quanticol.eu

### 3.2 Richer forms of interaction and adaptation

The current work on collective system modelling with stochastic process algebras has made limited use of functional rates to capture adaptation. For example, in the modelling of emergency egress a functional rate is used to represent how occupants might alter their planned route out of the building when they encounter congestion in a stairwell. As this illustrates, a functional rate is able to model adaptation in the form of adjusting the rate or probability of certain events to reflect the current situation. However this is only a limited form of adaptation.

In general, real collective adaptive systems, especially those with emergent behaviour, embody rich forms of interaction, often based on asynchronous communication. An example of this is the pheromone trail left by a social insects such as an ant. In this case the message (pheromone) left by one ant will affect the behaviour of another ant in the same location at a later time. Moreover, the patterns of communication, who can communicate with whom, may change over time according to the state of the system. Languages like SCEL offer these richer communication patterns [7]. In SCEL components include a knowledge store which can be manipulated by the component itself and other components. Communication can then be attribute-based, meaning that a message is sent to all components that have a given value for an attribute.

Again this differentiation through attributes poses a risk to fluid approximation. Accuracy in the fluid approximation relies on having a large enough subpopulation with shared characteristics. Allowing components to have distinct attribute values creates distinguishing features amongst the member of the subpopulations. Within the QUANTICOL project we are exploring ways to overcome these problems.

### 3.3 Extending model checking capabilities

Whilst many performance measures can be derived using the techniques of fluid rewards, more sophisticated interrogation of a model can be achieved through model checking. In stochastic model checking a suitably enhanced logic, CSL, specifies the query, and leads to a modification of the given CTMC. A naive approach based on fluid approximation would work directly with the vector field, but as this is deterministic this is amenable only to LTL model checking, and gives no indication of the inherent stochasticity in the system.

Recent work on *fluid model checking* develops an analogous approach for collective systems [3]. CSL properties related to a single component can be checked with respect to a population. In this approach the single component is left discrete and combined with a fluid approximation of the rest of the population, giving rise to a inhomogeneous time CTMC. This is then modified as in stochastic model checking, and solved numerically. Whilst effective, this approach can only be used to check the properties of one element of a population. In an alternative approach, based on a central limit approximation, the fraction of a population that satisfies a property expressed as a one-clock deterministic timed automaton can be checked [5]. Future work will seek to extend these to find scalable approaches to model checking global properties of collective systems.

## 4 Conclusions

Collective Adaptive Systems are an interesting and challenging class of systems to design and construct. Their role within infrastructure, such as within smart cities, make it essential that quantitive aspects of behaviour are taken into consideration, as well as functional correctness. Fluid approximation based analysis offers hope for scalable quantitative analysis techniques, but there remain many interesting and challenging problems to be solved.

## Acknowledgement

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

## References

1. M. Ajmone Marsan, G. Conte and G. Balbo. A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. In ACM Transactions on Computer Systems 2(2), pp. 93–122, 1984.
2. M. Bernardo and R. Gorrieri. A Tutorial on EMPA: A Theory of Concurrent Processes with Nondeterminism, Priorities, Probabilities and Time. In Theoretical Computer Science 202(1–2), pp. 1–54, 1998.
3. L. Bortolussi and J. Hillston, Checking Individual Agent Behaviours in Markov Population Models by Fluid Approximation. In Formal Methods for Dynamical Systems - 13th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2013, LNCS 7938, pp. 113–149, 2013.
4. L. Bortolussi, J. Hillston, D. Latella and M. Massink. Continuous approximation of collective system behaviour: A tutorial. In Performance Evaluation 70(5), pp. 317–349, 2013.
5. L. Bortolussi and R. Lanciani. Central Limit Approximation for Stochastic Model Checking. In Proceedings of Quantitative Evaluation of Systems - 10th International Conference, QEST 2013, LNCS 8054, pp. 123–138, 2013.
6. J.T. Bradley, S.T. Gilmore and J. Hillston. Analysing distributed Internet worm attacks using continuous state-space approximation of process algebra models. In Journal of Computer System Science 74(6), pp. 1013–1032, 2008.
7. R. De Nicola, G. Ferrari, M. Loreti, R. Pugliese. A Language-Based Approach to Autonomic Computing. in Formal Methods for Components and Objects, 10th International Symposium, FMCO 2011, Revised selected paper, LNCS 7542, pp. 25–48, 2011.
8. M.C. Guenther and J.T. Bradley. Mean-field analysis of data flows in wireless sensor networks. In ACM/SPEC International Conference on Performance Engineering, ICPE'13, pp. 51–62, 2013.
9. R.A. Hayden and J.T. Bradley. A Fluid Analysis Framework for a Markovian Process Algebra. Theoretical Computer Science 411(22-24), pp. 2260–2297, 2010.
10. R.A. Hayden, A. Stefanek and J.T. Bradley. Fluid Computation of Passage-time Distributions in Large Markov Models. In Theoretical Computer Science 413(1), pp. 106-141, 2012.
11. H. Hermanns. Interactive Markov Chains: The Quest for Quantified Quality. Springer, LNCS 2428, 2002.

12. J. Hillston. A Compositional Approach to Performance Modelling. Cambridge University Press, 2005.
13. J.Hillston. Fluid Flow Approximation of PEPA Models. In 2nd Int. Conf. on the Quantitative Evaluaiton of Systems (QEST 2005), pp. 33–43, 2005.
14. J. Hillston and L. Kloul. Formal techniques for performance analysis: blending SAN and PEPA. In Formal Aspects of Computing 19(1), pp. 3–33, 2007.
15. J. Hillston, M. Tribastone and S. Gilmore. Stochastic Process Algebras: From Individuals to Populations. In The Computer Journal 55(7), pp. 866–881, 2012.
16. M.Z. Kwiatkowska, G. Norman and D. Parker. PRISM: probabilistic model checking for performance and reliability analysis. In SIGMETRICS Performance Evaluation Review 36(4), pp. 40–45, 2009.
17. M. Massink, M. Brambilla, D. Latella, M. Dorigo and M. Birattari. On the use of Bio-PEPA for modelling and analysing collective behaviours in swarm robotics. In Swarm Intelligence 7(2–3), pp. 20–228, 2013.
18. M. Massink, D. Latella, A. Bracciali, M.D. Harrison and J. Hillston. Scalable context-dependent analysis of emergency egress models, In Formal Aspects of Computing 24(2), pp. 267–302, 2012.
19. M. Massink, M.D. Harrison and D. Latella. Scalable analysis of collective behaviour in smart service systems. In Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), pp. 1173–1180, 2010.
20. R.Milner. The Space and Motion of Communicating Agents. Cambridge University Press, 2009.
21. M. Tribastone, J. Ding, S. Gilmore and J. Hillston. Fluid Rewards for a Stochastic Process Algebra. In IEEE Transactions on Software Engineering, 38(4), pp. 861–874, 2012.
22. M. Tribastone, J. Hillston and S. Gilmore. Scalable Differential Analysis of Process Algebra Models. In IEEE Transactions on Software Engineering, 38(1), pp. 205–219, 2012.