Quantitative Analysis of Collective Adaptive Systems

Citation for published version:
Hillston, J 2016, Quantitative Analysis of Collective Adaptive Systems. in Perspectives of System Informatics: 10th International Andrei Ershov Informatics Conference, PSI 2015, in Memory of Helmut Veith, Kazan and Innopolis, Russia, August 24-27, 2015, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9609, Springer International Publishing, pp. 1-5, 10th International Andrei Ershov Informatics Conference, Kazan, Russian Federation, 24/08/15. DOI: 10.1007/978-3-319-41579-6_1

Digital Object Identifier (DOI):
10.1007/978-3-319-41579-6_1

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Perspectives of System Informatics

Quantitative Analysis of Collective Adaptive Systems

Jane Hillston

LFCS, School of Informatics, University of Edinburgh
jane.hillston@ed.ac.uk
http://www.quanticol.eu

1 Introduction

Quantitative formal methods, such as stochastic process algebras, have been used for the last twenty years to support modelling of dynamic systems in order to investigate their performance. Application domains have ranged from computer and communication systems [1, 2], to intracellular signalling pathways in biological cells [3, 4]. Nevertheless this modelling approach is challenged by the demands of modelling modern collective adaptive systems, many of which have a strong spatial aspect, adding to the complexity of both the modelling and the analysis tasks.

In this talk I gave an introduction to formal quantitative analysis and the challenges of modelling collective adaptive systems, together with recent developments to address those challenges using the modelling language CARMA.

2 Quantitative Analysis

Performance analysis has a long tradition in computer and communication engineering dating back to the 1960s, as the dynamic behaviour of systems can sometimes be counter-intuitive and hard to predict without detailed mathematical models. Since many aspects of the system must be abstracted in order to construct tractable models, probability distributions are used to represent the variability within the timing characteristics of the system, for example, due to different data characteristics. Specifically continuous time Markov chains (CTMCs) were found to offer a good compromise between faithfulness and tractability. Initially queueing networks [5] were the dominant approach to capturing the conflict for resources which is often at the root of performance problems. From such descriptions it is easy to build a CTMC, typically with a simple birth-death process for each queue, but in many cases this is not even necessary as analytical solutions are known, circumventing the need for the explicit construction and analysis of the CTMC [6].

However, the advent of large distributed systems, in which multiple resources may be needed by processes simultaneously, led to the use of more flexible modelling frameworks such as stochastic Petri nets [7] and stochastic process algebras [8]. Stochastic process algebras are small textual description languages which represent a system as a number of interacting processes. At the basic level each process is a small CTMC capturing the ordering and (stochastic) timing of activities that the process may undertake. The construction of the model specifies how these processes are constrained to interact through shared activities. A large CTMC capturing the complete behaviour of the system can be automatically constructed from the stochastic process algebra description, allowing the modeller to focus on the higher level behaviour of the system rather than the underlying state space. Prime examples of stochastic process algebras include PEPA [8], EMPA [9] and IMC [10]. Note, however, that the system description and the underlying mathematical model (CTMC) are inherently discrete and this can pose significant challenges for representing large scale systems due to the problem of *state space explosion*. In particular, numerical solution to find the probability distribution over the state space becomes intractable, and stochastic simulation becomes computationally expensive, as the size of the system grows.

3 Challenges in Modelling Collective Adaptive Systems

Recent years have seen increasing interest in collective adaptive systems. Such systems, which appear in many natural scenarios such as the behaviour of social insects, are increasingly forming a paradigm for the construction of the software systems of the future. Collective adaptive systems (CAS) are seen to be comprised of a large number of interacting entities whose behaviour is based on their local perception, without access to global control or knowledge. Moreover entities are typically replicated to establish robustness to failure of individual entities whilst adaptivity provides robustness to changes at a higher level.

The global or emergent behaviour of such systems can be difficult to predict making it paramount that we develop adequate modelling formalisms to capture and reason about the behaviour at both local and global levels. The compositional nature of stochastic process algebras make them strong candidates for developing models at the local level, but the state space explosion problem places severe challenges on their analysis.

In the last decade there have been efforts to alleviate the problems of state space explosion for very large systems by the use of *fluid approximations* [11]. In this approach an approximation of the underlying discrete CTMC is constructed as a set of ordinary differential equations that capture the average behaviour of the system when a large population of entities is involved. Analysis techniques, such as stochastic model checking, have been adapted to work with this approximation [12].

Other significant challenges for stochastic process algebras when modelling CAS stem from the spatially distributed nature of the entities and the adaptation. The spatial aspect is important because entities are restricted to interact and communicate locally so capturing the relative positions of entities is crucial. Similarly, the ability of an entity to have a goal which guides changes in behaviour based on the information that it receives has not previously been considered in stochastic process algebras. To address these challenges we have developed a new process algebra-based language CARMA [13].

4 CARMA

CARMA has been designed specifically to represent systems developed according to the CAS paradigm [14]. The language offers a rich set of communication primitives, and permits exploiting attributes, captured in a store associated with each component, to enable attribute-based communication. For most CAS systems we anticipate that one of the attributes could be the location of the agent. Thus it is straightforward to model those systems in which, for example, there is a limited scope of communication or there is the restriction to only interact with components that are co-located, or where there is spatial heterogeneity in the behaviour of agents. The use of a store to explicitly capture a limited set of data associated with an entity is a compromise between full agent-based modelling and the approach of data abstraction that has previously been adopted in stochastic process algebra-based languages.

The rich set of communication primitives is one of the distinctive features of CARMA. Specifically, CARMA supports both unicast and broadcast communication, and permits locally synchronous, but globally asynchronous communication. This richness is important to take into account the spatially distributed nature of CAS, where agents may have only local awareness of the system, yet the design objectives and adaptation goals are often expressed in terms of global behaviour. Representing these patterns of communication in classical process algebras or traditional stochastic process algebras would be difficult, and would require the introduction of additional model components to represent buffers, queues and other communication structures.

Another key feature of CARMA is its distinct treatment of the *environment*. It should be stressed that although this is an entity explicitly introduced within our models, it is intended to represent something more pervasive and diffusive of the real system, which is abstracted within the modelling to be an entity which exercises influence and imposes constraints on the different agents in the system. For example, in a model of a smart transport system, the environment may have responsibility for determining the rate at which entities (buses, bikes, taxis etc) move through the city. However this should be recognised as an abstraction of the presence of other vehicles causing congestion which may impede the progress of the focus entities to a greater or lesser extent at different times of the day. The presence of an environment in the model does not imply the existence of centralised control in the system. The role of the environment is also related to the spatially distributed nature of CAS — we expect that the location *where* an agent is will have an effect on *what* an agent can do.

To summarise, in CARMA a system is composed of a *collective* of components that exist in an *environment*. Each component consists of a *process* and a *store* where the process captures the possible behaviours of the component in a similar manner to previous process algebras, whereas the store records the state of the component with respect to a number of *attributes*. These attributes, which can be thought of as enumerated types, allow the behaviour, including the communication partners, of a component to be dependent on its current state.

5 Future Perspectives

CAS present an interesting and challenging class of systems to design and construct, with many exciting prospects for future software systems as well as socio-technical systems, in which users themselves become entities in the system. Example areas of application include smart urban transport systems, smart energy networks and swarm robotics. The role of CAS within infrastructure systems, such as within smart cities, make it essential that quantitive aspects of behaviour, in addition to functional correctness, are taken into consideration during design, but the scale and complexity of these systems pose challenges both for model construction and model analysis. CARMA aims to address many of these challenges, supporting rich forms of interaction, using attributes to capture explicit locations and the environment to allow adaptivity. Moreover analysis techniques based on fluid approximation offer hope for scalable quantitative analysis techniques. However, to tackle the full range of behaviours which can occur within CAS extensions to classical fluid approximation techniques are needed. For example, recent work by Bortolussi [15] proves convergence results in terms of hybrid systems which take into account multi-scale behaviour with respect to time and/or populations.

Acknowledgement

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

References

1. Hermanns, H., Herzog, U., Katoen, J.: Process algebra for performance evaluation. Theor. Comput. Sci. **274**(1-2) (2002) 43–87
2. De Nicola, R., Latella, D., Massink, M.: Formal modeling and quantitative analysis of klaim-based mobile systems. In: Proc. of the 2005 ACM Symposium on Applied Computing (SAC), Santa Fe, New Mexico, USA, March 13-17, 2005, ACM (2005) 428–435
3. Priami, C.: Algorithmic systems biology. Commun. ACM **52**(5) (2009) 80–88
4. Ciocchetta, F., Hillston, J.: Bio-PEPA: A framework for the modelling and analysis of biological systems. Theoretical Computer Science **410**(33) (2009) 3065–3084
5. Kleinrock, L.: Sequential processing machines (S.P.M) analyzed with a queuing theory model. J. ACM **13**(2) (1966) 179–193
6. Baskett, F., Chandy, K.M., Muntz, R.R., Palacios, F.G.: Open, closed, and mixed networks of queues with different classes of customers. J. ACM **22**(2) (1975) 248–260
7. Marsan, M.A., Conte, G., Balbo, G.: A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems. ACM Trans. Comput. Syst. **2**(2) (1984) 93–122
8. Hillston, J.: A Compositional Approach to Performance Modelling. CUP (1995)
9. Bernardo, M., Gorrieri, R.: A Tutorial on EMPA: A Theory of Concurrent Processes with Nondeterminism, Priorities, Probabilities and Time. Theoretical Computer Science **202**(1-2) (1998) 1–54
10. Hermanns, H.: Interactive Markov Chains: The Quest for Quantified Quality. LNCS. Springer (202)
11. Hillston, J.: The benefits of sometimes not being discrete. In Baldan, P., Gorla, D., eds.: CONCUR 2014 - Concurrency Theory - 25th International Conference, CONCUR 2014, Rome, Italy, September 2-5, 2014. Proceedings. Volume 8704 of Lecture Notes in Computer Science., Springer (2014) 7–22
12. Bortolussi, L., Hillston, J.: Model checking single agent behaviours by fluid approximation. Inf. Comput. **242** (2015) 183–226

13. Hillston, J., Loreti, M.: Specification and analysis of open-ended systems with CARMA. In Weyns, D., Michel, F., eds.: Agent Environments for Multi-Agent Systems IV - 4th International Workshop, E4MAS 2014 - 10 Years Later, Paris, France, May 6, 2014, Revised Selected and Invited Papers. Volume 9068 of Lecture Notes in Computer Science., Springer (2014) 95–116
14. Bortolussi, L., De Nicola, R., Galpin, V., Gilmore, S., Hillston, J., Latella, D., Loreti, M., Massink, M.: CARMA: collective adaptive resource-sharing markovian agents. In Bertrand, N., Tribastone, M., eds.: Proceedings Thirteenth Workshop on Quantitative Aspects of Programming Languages and Systems, QAPL 2015, London, UK, 11th-12th April 2015. Volume 194 of EPTCS. (2015) 16–31
15. Bortolussi, L.: Hybrid behaviour of markov population models. CoRR **abs/1211.1643** (2012) to appear in Information and Computation 2016.