QUANTICOL

A Quantitative Approach to Management and Design of Collective and Adaptive Behaviours

quantiCOL

http://www.quanticol.eu

Internal Report 1.1

Linking language and mean field approximations

Revision: 1.1; 28 September 2015

Author(s): Luca Bortolussi (CNR), Jane Hillston (UEDIN), Michele Loreti (IMT)

Publication date: September 30, 2015

Funding Scheme: Small or medium scale focused research project (STREP)
Topic: ICT-2011 9.10: FET-Proactive 'Fundamentals of Collective Adaptive Systems' (FOCAS)
Project number: 600708

Coordinator: Jane Hillston (UEDIN)
e-mail: Jane.Hillston@ed.ac.uk
Fax: +44 131 651 1426

Part. no.	Participant organisation name	Acronym	Country
1 (Coord.)	University of Edinburgh	UEDIN	UK
2	Consiglio Nazionale delle Ricerche – Istituto di Scienza e Tecnologie della Informazione "A. Faedo"	CNR	Italy
3	Ludwig-Maximilians-Universität München	LMU	Germany
4	Ecole Polytechnique Fédérale de Lausanne	EPFL	Switzerland
5	IMT Lucca	IMT	Italy
6	University of Southampton	SOTON	UK
7	Institut National de Recherche en Informatique et en Automatique	INRIA	France

COOPERATION

Abstract

The aim of this document is to identify the first steps towards integrating the results from Task1.1 in Work Package 1 with the language development work that is being carried out in Work Package 4. Specifically we consider how the emerging features of the CARMA language can be used to take advantage of the scalable analysis techniques developed in Task1.1. Moreover, we also assess what changes might be required for CARMA in order to link with the results on uncertain CTMCs, and their approximation by differential inclusions in Task1.1.

Contents

1 Introduction

The CARMA language will be the main vehicle for delivery of the theoretical results that are established in the QUANTICOL project [CNH+15, BNG+15]. Thus it is important that we ensure that as CARMA is developed and enshrined in software, it has the necessary features to allow those results to be exploited when appropriate. CARMA is a very rich language and it is therefore likely that in some cases restrictions will need to be placed on the language in order to ensure that the correct underlying mathematical representation is obtained to allow a particular result to be used. Where possible, we will aim for these restrictions to be identified by static analysis, implemented in software and integrated into the design workflow and analysis pathway being developed in Task 4.3. The objective of the current document is to take the first steps towards this integration of the theoretical results and the practical modelling support to be developed through CARMA.

The rest of the document is organised as follows. In Section 2, we consider the results on mean field and hybrid limits that have been developed in Task1.1 and consider the steps that are needed in order to take advantage of these through establishing a semantics of CARMA in terms of population Markov processes (PMP). In Deliverable D1.2 and associated papers we explain how the limit of a PMP as the population tends to infinity is a hybrid mean field limit [BGH15, Bor15]. Once a PMP semantics for CARMA is developed, in order to take advantage of this result to form an efficient scalable representation suitable for the quantitative analysis of models of large scale collective adaptive systems (CAS), it will only remain to establish for any particular model that it scales appropriately. Further results of Task1.1 [BGH15], reported in Deliverable 1.2, consider models in which not all parameter values are known, leading to the definition of uncertain and imprecise CTMC models. In Section 3 we present how the present syntax of CARMA could be extended to allow the description of such uncertainty, which would then be mapped to an uncertain CTMC, taking advantage of fluid limit results in terms of differential hulls, as discussed in [BGH15, TT15]. Finally in Section 4 we discuss the future work and relationship with other work packages.

2 Models with mean field and hybrid limits

In previous work [TGH12, Hil05] we have established that for a process algebra with a semantics in terms of a continuous time Markov chain (CTMC) it is feasible to establish a link between mean field limit results and the semantics of the language, meaning that it is possible to equip the language with a mean field semantics and automatically derive the limit process. This means that the suitability of the mean field approximation can be established once for the language rather than having to be established on a model-by-model basis. Within the QUANTICOL project this approach has been extended to PALOMA, a stochastic process algebra in which agents have explicit locations [FH14], but this work was based on the same mean field limit derived from Kurtz's original work [Kur81] for CTMC models. The recent work in Task1.1 has extended this approach to consider richer mathematical processes resulting in hybrid limits [BGH15, Bor15], allowing more expressive modelling languages, such as CARMA, to be given a mean field style semantics. Nevertheless, given the power of the CARMA, work remains to be done to identify just how to link the language and the mean field/hybrid limit results. This report represents the first steps along that path.

CARMA models are highly structured, consisting of a *collective* of *components* operating in a *environment*, with the behaviour of each component defined by one or more *processes* integrated with a data *store* made up of *attributes* (see Deliverable 4.2 for more details). In contrast the target mathematical models which facilitate the application of the mean field and hybrid limits are much less structured so the first step is to decide how to map the language structures to the mathematical ones.

2.1 Models without a global store

As we will present below, if a CARMA model does not have a global store, a population CTMC (PCTMC) can be constructed and given a mean field semantics following from Kurtz's original result since the resulting mathematical model will be without discontinuities. The *populations* arise from agents who have distinct patterns of behaviour, from their own definitions or from their interactions with other agents/populations. We assume that the identity of individual agents does not influence the dynamics, hence we just need to count how many agents are in each state. Thus, we introduce integer-valued counting variables $\mathbf{X} = (X_1, \ldots, X_n)$, with n the number of different agent states. These variables take values in a subset $S \subseteq \mathbb{N}^n$, which is the state space of the model. The dynamics of Markov population models is described, according to [BHLM13, Bor15], by a collection $\mathcal{T}$ of transition classes, each consisting in a tuple $\eta = (a, \phi(\mathbf{X}), \nu, f(\mathbf{X}))$. Specifically, a is the transition label, ν is the n-dimensional update vector, giving the net change in the number of agents for each state, $f(\mathbf{X})$ is the population dependent rate of the transition, encoding its average frequency, and $\phi(\mathbf{X})$ is a state dependent Boolean predicate such the the transition is active only when it evaluates to true.

A Markov population model is therefore defined by a vector of counting variables $\mathbf{X}$, a set of transition classes $\mathcal{T}$, and an initial state $\mathbf{x_0} \in S$. From this description, it is straightforward to derive the underlying CTMC, denoted by $\mathbf{X}(t)$, see [BHLM13] for further details. The mean field limit arises when we consider the behaviour of the model when the size of the system N (usually the initial population of agents) tends to infinity. Classic mean field limits show that a Markov population model, with a proper scaling of rates and with variables divided by N, converges to a deterministic limit given by the solution of a set of differential equations.

If we consider the collective $\mathcal{C}$ of a CARMA model, setting aside for the moment the environment, we can see how to map to this PCTMC framework.

1. The behaviour within each component is mapped to a behavioural agent, which has one state for each state of the *processes* within the component. It is important to note that in this step any interleaving of the behaviour of the internal processes within a component is flattened, so that essentially each composition of processes is reduced to a single automaton capturing the behavioural states of the component. We denote this set of states $ds(C)$, the derivative set of the component C.

2. Secondly, the possible states of the store must also be taken into account. In CARMA we assume that the attributes have enumerated types so that we can define a finite set $D(C)$ which is the attribute domain of component C; this is the set of all possible values that the different attributes of C can take. We list these states as $\{v_1, \ldots, v_k\}$, and treat them as single entities, rather than vectors of attribute values.

3. Thirdly, we combine the behavioural agent with the attribute values to form an agent type $A_{C,P,v}$ where $C \in \mathcal{C}$ is a component in the collective, $P \in ds(C)$ is a behavioural state of C and $v \in D(C)$ is a value assignment to attributes of C.

These steps identify the agent types within the PCTMC model. In order to build the PCTMC we must construct a population vector which counts the number of each agent type which are present within the model. We can define the population vector as $\mathbf{X} = (X_{C,P,\nu})$ where $X_{C,P,\nu}$ is the number of instances of $A_{C,P,v}$ present in the model. We assume that we have a definition of component types $\mathcal{C}$, which are all the distinct components in the model. Two components are the same if they have the same derivative set and the same attribute domain. Given a CARMA model, we assume a function *count* associating to a parallel composition of components C a population vector $\mathbf{X} = count(C)$ counting how many components of kind C are in local state $(P, \gamma \mapsto \nu)$.

As explained above, the dynamics of the PCTMC are defined in terms of transition classes, with each transition having a transition label, an update vector and a rate function. For a model derived from CARMA the transition label will be given by the action name of the action which brings about

the transition. We postpone consideration of the rate function as that is defined in the environment, and focus here on the definition of the update vector. We define $\mathbf{e}_{C,P,\nu}$ to be the indicator vector of component C, P, ν within $\mathbf{X}$.

In order to construct the update vector we must identify the possible participants in a transition, say α, and construct an appropriate adjustment to the agent counts according to which agent types are involved, using the indicator vectors $\mathbf{e}_{C,P,\nu}$. Consider a broadcast action α^*. We again proceed in a number of steps:

1. We distinguish between those agents which can initiate the action — *sender agents* who are derived from a process which has this action as output — and those that will respond to it — *receiver agents* who are derived from a process which as this action as input. We will call $sender_b(\alpha)$ the set of all possible senders, i.e. those components in a state P and with store ν, such that P can perform a broadcast output α^*.

2. For any other agent tuple (C', P', ν'), we place it in $receiver_b(\alpha, C, P, \nu)$ if and only if all the following conditions hold

   - P' can perform a broadcast input of type α
   - $\pi_{out}[\nu']$ is true
   - $\pi_{in}[\nu]$ is true.

   where π_{out} is the output guard predicate for α^* and π_{in} is the guard in the predicate of the input action.

3. This gives the elements of the population vector which will *potentially* change when the action is completed. Only the agent which will send the broadcast is certain to be updated, with an update represented by $\mathbf{e}_{C,P,\nu}$. Otherwise, there are two aspects which make this interaction only potential.

   - Firstly, for the sender the subsequent agent/state may not be deterministic. The action in the sender may have an update function which includes a distribution over possible attribute values. Thus there is a probability distribution over potential updates resulting from the sender. This is represented in the PCTMC framework by a random update, which selects each possible update vector $\mathbf{e}_{C,P,\nu}$ proportionally to the probability $p_{\sigma(\nu)}$ with which it will occur.
   - Secondly, the reception of a message has an associated probability (defined in the environment through the evolution rule), meaning that there is a binomial distribution over which of the potential receivers will actually receive the message and change state (reducing the count). Within the set of successful receivers, there will again be a distribution over attribute updates leading to a more structured random update (increasing the count of some agents).

The situation with a unicast action is similar but slightly simpler since only one partner is chosen to receive each output, but we still must reflect the uniform nature of that choice in the updates which are increasing the count. This will result again in a random update vector, where each potential receiver class will be selected with a probability proportional to its number of occurrences in the model.

The final step in defining each transition class is to define the rate of the transition. For this we must bring the environment back into consideration, because it is within the evolution rule of the environment that the rates and probabilities associated with each action type are defined. According to the evolution rule these functions may depend on the current state of the system, but since that is fully captured by $\mathbf{X}$ this does not present any difficulties.

If the environment does not have a global store and the size of the populations of the agents derived following the outline above scale appropriately, then we can be confident that the PCTMC which is

constructed in this way will be amenable to a fluid limit based on Kurtz's original result [Kur81], and lead to a mean field approximation for a CARMA model.

2.2 Models with a global store

When a model has a global store we need to consider two possibilities with respect to how the values in the global store change as the population of agents (system size N) tends to infinity. The state of the global store will be represented by a vector of variables $\mathbf{A}$, using a boolean string encoding of each attribute (i.e. an attribute with k states will be represented by k boolean variables, with variable i equal to one if and only if the value of the attribute is the i-th one), in order to comply with PCTMC formalism. However, here we consider the vector to consist of two sub vectors $\mathbf{A} = (\mathbf{A}_f, \mathbf{A}_s)$. $\mathbf{A}_f$ represents the *fast* element of the environment, i.e. those which can be modified by transitions whole speed scales as the populations of the system scales. In the mathematical encoding, such variables will be removed by invoking a time-scale separation argument, following the idea of [BLB08]: in the limit, their state will change so fast that they will reach instantaneously their steady state distribution, assumed unique (which can be checked automatically by a graph-based analysis of the transitions between different states in the global environment). In contrast, $\mathbf{A}_s$ represents the *slow* element of the environment, i.e. those transitions which are unaffected by the number of agents in the model. In the mathematical encoding these variable will give rise to population counts which remain discrete in the limit giving rise to a hybrid behaviour, taking advantage of the results in [Bor15].

Now we consider a Markov Population Model (MPM) in which the counting variables $\mathbf{X}$ can be partitioned into to classes: $\mathbf{Z}$, the discrete variables (corresponding to the slow attributes in the global store) and $\mathbf{Y}$, the continuous variables (corresponding to the fast attributes in the global store and the agents arising from the collective). Similarly we distinguish the transitions which occur in the system.

- **Continuous transitions** modify only continuous variables $\mathbf{Y}$ and their rate will be proportional to the system size N.
- **Discrete stochastic transitions with no jumps** modify both discrete and continuous variables with a rate that is independent of N, but we assume that in the case of continuous variables the corresponding entry in the update vector is independent of N and so in the limit (with normalised variables) the net effect is zero.
- **Discrete stochastic transitions with jumps** modify both discrete and continuous variables, again with a rate that is independent of N. But in this case the update associated with a continuous variable can be proportional to N, meaning that in the limit such a transition induces a discontinuous jump in the normalised continuous variables $\hat{\mathbf{Y}}$.

Once variables and transitions are partitioned into classes as described above, we can consider the behaviour of the MPM as the system size N tends to infinity, obtaining a stochastic hybrid system, which may be defined in terms of a Piecewise Deterministic Markov Process [Dav93]. Within this hybrid system there is one mode for each possible state that the discrete variables $\mathbf{Z}$ can take, while continuous variables $\hat{\mathbf{Y}}$ define its continuous state space, and execution of the system involves alternating phases of continuous evolution and discrete jumps. In terms of CARMA this means that the discrete modes of the limit process will correspond to the elements of $\mathbf{A}_s$ and there will be set of ODEs describing the behaviour of the agents and the fast environment variables for each of these modes.

In order to form such a hybrid limit from a CARMA model it must be possible to partition the variables and transitions in the way that is described above. This may place restrictions on the form of evolution rule which can be included in the environment of a model. In the CARMA definition the evolution rule is a function which, depending on the *current time*, on the global store and on the current state of the collective (i.e., on the configurations of each component in the collective) returns a tuple of functions $\varepsilon = \langle \mu_p, \mu_r, \mu_u \rangle$ known as the *evaluation context* where ACT = ACTTYPE $\cup \{\alpha^\star | \alpha \in$ ACTTYPE$\}$ and:

- $\mu_p : \Gamma \times \Gamma \times \text{Act} \to [0,1]$, $\mu_p(\gamma_s, \gamma_r, \alpha)$ expresses the probability that a component with store γ_r can receive a message from a component with store γ_s when α is executed;
- $\mu_r : \Gamma \times \text{Act} \to \mathbb{R}_{\geq 0}$, $\mu_r(\gamma, \alpha)$ computes the execution rate of action α executed at a component with store γ;
- $\mu_u : \Gamma \times \text{Act} \to \Sigma \times \text{Col}$, $\mu_u(\gamma, \alpha)$ determines the updates on the environment (global store and collective) induced by the execution of action α at a component with store γ.

We can identify a number of restrictions to ensure that an appropriate MPM is formed from the Carma model.

- The first restriction that we must impose is that the evolution rule should not depend on the current time. Dependence on the current time, as explained in Deliverable 4.2 [CNH+15], gives rise to a semantics which is based on a time-inhomogeneous continuous time Markov chain, rather than a CTMC, and consequently the limit results established in Task1.1 and previously (in their current form) would not apply. Relaxation of these constraints requires a preliminary theoretical investigation of mean field limits for time-inhomogeneous models.
- Valid continuous transitions will arise if the rate functions associated with the actions in the Carma model do not affect any slow variables in the environment. This means that for an action type α in a component with local store γ, $\mu_u(\gamma, \alpha) \cap \mathbf{A}_s = \emptyset$[1]. Also $O(\mu_r(\gamma, \alpha)) = N$ (the rate scales with the population size).
- If we impose that whenever the update function μ_u of an action type α does modify a slow environment variable, i.e. $\mu_u(\gamma, \alpha) \cap \mathbf{A}_s \neq \emptyset$, then it does not modify any counting variables arising from agents or fast environment variables, i.e. all α actions have an empty update and $\mu_u(\gamma, \alpha) \cap \mathbf{A}_f = \emptyset$. Moreover the rate of action α must be independent of N, the function $\mu_r(\gamma, \alpha)$ does not depend on any populations within the model. These conditions are stronger than needed but will guarantee discrete stochastic transitions with no jumps of the valid form, and should be easy to check in the syntax of the model.
- In order to give rise to discrete stochastic transitions with jumps, we would relax the previous condition to allow an action type α to have an update function with modifies both fast and slow environment variables, i.e. $\mu_u(\gamma, \alpha) \cap \mathbf{A}_s \neq \emptyset \neq \mu_u(\gamma, \alpha) \cap \mathbf{A}_f$. In this case we would then need to impose that either
  - the rate function, $\mu_r(\gamma, \alpha)$ is independent of N (does not depend on any populations) and that all updates arising from α either in the prefix expressions (affecting the agent counting variables) or in the environment update (affecting the fast environment variables) are also independent of N leading to discrete stochastic transitions with no jumps; or
  - the rate function, $\mu_r(\gamma, \alpha)$ is independent of N and all updates arising from α either in the prefix expressions or the environment update are proportional to the system size N leading to discrete stochastic transitions with jumps.

We believe that this framework provides a sound basis for the development of a hybrid semantics of Carma models and their efficient scalable analysis based on the hybrid limit results of Task 1.1.

3 Models with parameter uncertainty

From the prospective of language support the key consideration for Carma specifications which are able to take advantage of the results developed on parameter uncertainty in Task 1.1, is modifying the

[1] with slight abuse of notation

syntax in order to capture the uncertainty. However, as shown in the work on introducing uncertainty to Bio-PEPA [GHMS14], it requires only minor modification to change the concrete specification of rates to a distribution over possible values. Presently the rates of all actions are defined within the environment of a CARMA model within the evolution rule, as explained above. In particular, within the CARMA specification language the syntax of a rate definition is:

```
rate { ···
    [guard_i] act_i : expr_i; ···
    default : expr;
}
```

where $expr_i$ is a function call, or an expression, returning a single real number which is taken as the rate of the exponential distribution associated with action act_i. In order to support imprecise or uncertain models rather than a single number we need these single values to be replaced by a in interval of possible values, e.g.:

```
rate { ···
    [guard_i] act_i : [expr_{i.lower}, expr_{i.upper}] ; ···
    default : expr;
}
```

where $expr_{i.lower}$ and $expr_{i.upper}$ represent the lower and the upper bound of the rate of act_i.

Modifications to the language along these lines will allow us to link with the work on uncertain and imprecise processes developed in Task1.1 and collected in the technical report [BG15]. In the case of the uncertain scenario, the rate to be used will be resolved before the model is passed to the solver, as one value will be chosen prior to model execution. Thus we can envisage a preprocessing step that resolves the uncertainty, reducing the interval rate specification to a single value and this concrete CARMA model is executed. No modification to the solver or semantics is required.

In contrast, in the case of an imprecise scenario, the value to be used may vary throughout the execution of the model, and consequently a semantics will be needed which maps the CARMA model to a Continuous-Time Markov Decision Process (CT-MDP) with continuous actions in the case of discrete event simulation, or fluid semantics in terms of imprecise drift captured by differential inclusions. However, this will be a second step after first developing the fluid/hybrid semantics of CARMA models with concrete rates.

4 Conclusions

In this report we have considered how the recent developments in Work Package 1, particularly those related to mean field and hybrid approximations, can be linked to the ongoing work on the language CARMA. To conclude we briefly outline the future work suggested by this report and discuss the relationship between this aspect of Work Package 1 and other work packages in the project.

4.1 Roadmap for future work and relation with other work packages

In the remaining period of the project we will work to make the ideas explained within this report more concrete, through collaboration between the CARMA language designers in Work Package 4, and the developers of the scalable analysis techniques based on mean field and fluid limits. We anticipate that the syntactic conditions outlined in Section 2.2 for models with a global store can be refined and supported by a formal mapping.

The key relation for the ideas presented here is with the work on language design and semantics in Tasks 4.1 and 4.2 in Work Package 4 and the implementation of the tool support for the language in Task 5.3 in Work Package 5. The mean field and hybrid semantics will allow the tool to offer scalable approximations for CARMA models. These will be subject to some restrictions on the models which can be treated in this way, as discussed in Section 2, and these restrictions will be integrated into the work on design workflow and analysis pathway being developed in Task 4.3 in order to allow

suitable models to be identified automatically. There is also scope for some interaction with the model reduction techniques being developed in Task 3.2 of Work Package 3.

References

[BG15] L. Bortolussi and N. Gast. Mean field approximation of imprecise population processes. *QUANTICOL Technical Report*, 2015.

[BGH15] L. Bortolussi, N. Gast, and J. Hillston. A framework for hybrid limits under uncertainty, 2015. QUANTICOL Deliverable D1.2.

[BHLM13] Luca Bortolussi, Jane Hillston, Diego Latella, and Mieke Massink. Continuous approximation of collective systems behaviour: a tutorial. *Performance Evaluation*, 2013.

[BLB08] Michel Benaim and Jean-Yves Le Boudec. A class of mean field interaction models for computer and communication systems. *Performance Evaluation*, 65(11):823–838, 2008.

[BNG+15] Luca Bortolussi, Rocco De Nicola, Vasti Galpin, Stephen Gilmore, Jane Hillston, Diego Latella, Michele Loreti, and Mieke Massink. CARMA: Collective adaptive resource-sharing markovian agents. In *Proc. of the Workshop on Quantitative Analysis of Programming Languages 2015*, 2015. to appear.

[Bor15] Luca Bortolussi. Hybrid behaviour of Markov Population Models. *Information and Computation*, accepted, 2015.

[CNH+15] V. Ciancia, R. De Nicola, J. Hillston, D. Latella, M. Loreti, and M. Massink. CAS-SCEL semantics and implementation, 2015. QUANTICOL Deliverable D4.2.

[Dav93] M. H. A. Davis. *Markov Models and Optimization*. Chapman & Hall, 1993.

[FH14] C. Feng and J. Hillston. PALOMA: A process algebra for located markovian agents. In *Proceedings of QEST 2014*, volume 8657 of *Lecture Notes in Computer Science*, pages 265–280. Springer, 2014.

[GHMS14] A. Georgoulas, J. Hillston, D. Milios, and G. Sanguinetti. Probabilistic programming process algebra. In *Quantitative Evaluation of Systems - 11th International Conference, QEST 2014, Florence, Italy, September 8-10, 2014. Proceedings*, volume 8657 of *Lecture Notes in Computer Science*, pages 249–264. Springer, 2014.

[Hil05] J. Hillston. Fluid flow approximation of PEPA models. In *2nd International Conference on the Quantitative Evaluation of Systems*, pages 33–42. IEEE, 2005.

[Kur81] T.G. Kurtz. *Approximation of population processes*. SIAM, 1981.

[TGH12] Mirco Tribastone, Stephen Gilmore, and Jane Hillston. Scalable differential analysis of process algebra models. *IEEE Transactions on Software Engineering*, 38(1):205–219, 2012.

[TT15] Max Tschaikowski and Mirco Tribastone. Approximate reduction of heterogenous nonlinear models with differential hulls. *IEEE Transactions on Automatic Control*, 2015.