QUANTICOL

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

quantiCOL

http://www.quanticol.eu

D3.1

Foundations of scalable verification for stochastic logics

Revision: 1.0; March 24, 2014

Author(s): Mieke Massink (CNR) – Luca Bortolussi (CNR), Vincenzo Ciancia (CNR), Jane Hillston (UEDIN), Alberto Lluch-Lafuente (IMT), Diego Latella (CNR), Michele Loreti (IMT), Daniël Reijsbergen (UEDIN), Andrea Vandin (SOTON)

Due date of deliverable: Month 12 (March 2014)

Actual submission date: March 31, 2014

Nature: R. Dissemination level: PU

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

Executive Summary

This deliverable reports on the study of the theoretical foundations of scalable model checking approaches, including those based on mean-field and fluid flow techniques, addressing the first phase of Task 3.1. Model checking has been widely recognised as a powerful approach to the automatic verification of concurrent and distributed systems. It consists of an efficient procedure that, given an abstract model of the behaviour of the system, decides whether that model satisfies properties of interest, typically expressed using a form of temporal logic. Despite their success, scalability of model checking procedures has always been a concern due to the potential combinatorial explosion of the state space that needs to be searched. This is particularly an issue when analysing large collective adaptive systems that typically consist of a large number of independent, communicating objects.

This deliverable reports on the theoretical foundations of several novel scalable model checking approaches that have been developed during the first year of the QUANTICOL project to address the challenge of scalability. A first approach concerns Fluid Model Checking. This approach exploits fluid flow approximations and fast simulation techniques in a global model checking algorithm to verify continuous stochastic logic properties of an individual object, or a few objects, in the context of large population models. The approach has been extended considering steady state properties and a method has been developed to lift local properties to global collective ones exploiting the central limit theorem. The second approach concerns mean-field fast probabilistic model checking. This approach exploits mean-field approximation and fast simulation in a discrete time probabilistic setting. Furthermore, it combines the above mentioned techniques with on-the-fly model checking techniques. The asymptotic correctness of the approach is outlined and the use of the prototype implementation of the algorithm, developed during the first year of the QUANTICOL Project, is briefly illustrated on a variant of the bike-sharing case study. A third approach concerns a novel and efficient statistical model checking technique and tool to deal with uncertainty in the values of model parameters in a statistically sound way. The approach is based also on recent advances in machine learning and pattern recognition.

Finally, we report on a literature study that was conducted on spatial logics and that is complementing the work on spatial representations in Work Package 2, in preparation for the future extension of the scalable model checking techniques addressing properties of spatially inhomogeneous collective adaptive systems.

Contents

1 Introduction

This deliverable reports on the study of the theoretical foundations of scalable model checking approaches, including those based on mean-field and fluid flow techniques, addressing the first phase of Task 3.1 of Work Package 3. We refer to Section 2 of Deliverable 1.1 for a brief introduction to mean-field and fluid techniques. Model checking has been widely recognised as a powerful approach to the automatic verification of concurrent and distributed systems. It consists of an efficient procedure that, given an abstract model $\mathcal{M}$ of the system, decides whether $\mathcal{M}$ satisfies a logical formula Φ, typically drawn from a temporal logic. Despite the success of model checking procedures, their scalability have always been a concern due to the potential combinatorial explosion of the state space that needs to be searched.

Traditionally, model checking approaches are divided into two broad categories: *global* approaches that determine the set of *all* states in $\mathcal{M}$ that satisfy Φ, and *local* approaches that, given a state s in $\mathcal{M}$, determine whether s satisfies Φ [CVWY92, BCG95]. Global symbolic model checking algorithms are popular because of their computational efficiency and are used in many model checkers, both in a qualitative setting (see e.g. [CES86]) and in a stochastic setting (see e.g. [BHHK03, KNP04]). The set of states that satisfy a formula is constructed recursively in a *bottom-up* fashion following the syntactic structure of the formula. Depending on the particular formula to verify, usually the underlying model can be reduced to fewer states before the algorithm is applied. Moreover, as is shown e.g. in [BHHK03] for stochastic model checking, the model checking algorithm can be reduced to combinations of existing well-known and optimised algorithms for CTMCs such as transient analysis.

Local model checking algorithms have been proposed to mitigate the state space explosion problem using a so-called 'on-the-fly' approach (see e.g. [CVWY92, BCG95, Hol04, GM11]). On-the-fly algorithms are following a *top-down* approach that does not require global knowledge of the complete state space. For each state that is encountered, starting from a given state, the outgoing transitions are followed to adjacent states, constructing step-by-step local knowledge of the state space until it is possible to decide whether the given state satisfies the formula. For qualitative model checking, local model checking algorithms have been shown to have the same worst-case complexity as the best existing global procedures for the above mentioned logics. However, in practice, they have better performance when only a subset of the system states need to be analysed to determine whether a system satisfies a formula. Furthermore, local model checking may still provide some results in the case of systems with a very large or even infinite state space where global model checking approaches would be impossible to use.

Besides global and local model checking techniques, recently a third model checking approach has been proposed, in particular for stochastic models, which is based on statistical analysis. The idea behind statistical model checking is to generate a sufficient number of independent finite simulation trajectories based on the model, check for which of the trajectories the property of interest holds and use this statistical information to estimate the satisfaction probability of the property [YS02, YS06, YKNP04, JCL$^+$09, ZPC10]. Even though statistical model checking does not require the generation of the state space of the model, the complexity of the approach nevertheless increases with the number of objects in the model. Unfortunately, all the above mentioned approaches essentially fail to solve the challenging scalability problems of collective adaptive systems (CAS) such as gossip protocols [CLR09], self-organised collective decision making [MFS$^+$11], computer epidemics [BGH08] and foreseen smart urban transportation systems and decentralised control strategies for smart grids like those contemplated by the QUANTICOL Project.

This deliverable reports on the study of the theoretical foundations of novel scalable model checking approaches that exploit fluid flow and mean-field approximations in combination with fast simulation. It also reports on a novel approach to statistical model checking and on a preliminary literature study on logics for space and time.

The proposed model checking approaches involve a basic stochastic logic and stochastic modelling language and the development of innovative and efficient fluid model checking and mean-field model

checking techniques addressing local, global and mixed stochastic properties of CAS and of individual components that are evolving in the context of a CAS. In particular, in Section 2 Fluid Model Checking is introduced that exploits fluid flow approximations and fast simulation techniques in a global model checking algorithm to verify continuous stochastic logic properties of an individual object, or a few objects, in the context of large population models. The approach is extended considering steady state properties and a method has been developed to lift local properties to global collective ones exploiting the central limit theorem. In Section 3 a mean-field probabilistic model checking approach is introduced that exploits mean-field approximation and fast simulation in a discrete time probabilistic setting and using an on-the-fly model checking algorithm. Asymptotic correctness of the approach was proven and a prototype implementation of the algorithm has been developed and applied to several case studies. Section 4 describes a novel and efficient statistical model checking approach and tool to deal with uncertainty in the values of the model parameters in a statistically sound way. The method is also based on recent advances in machine learning and pattern recognition. Finally, in Section 5 a literature study of spatial logics is presented, complementing the work on spatial representations in Deliverable 2.1. This work has been conducted in preparation for the future extension of the scalable model checking techniques addressing properties of spatially inhomogeneous collective adaptive systems. Section 6 summarises the main results and presents a work plan for the next reporting period.

2 Stochastic logic and model checking techniques that exploit fluid flow techniques

2.1 State of the art/base line before start of project

Fluid flow approximation [Kur81, LMM07, BL08, DN08] (see also [BHLM13] for a tutorial introduction) has proved a very successful technique to analyse large population models. In recent years, it has been applied to models described by stochastic process algebras, resulting in a new analysis procedure, which worked very well to approximate system averages and rewards [Hil05, TDGH12, TGH12, HB10]. Stochastic population models can also be analysed by model checking methods, either exact [KNP04] or statistical [YKNP04]. These tools, however, suffer from state space explosion[1], and quickly become inefficient not only for realistic systems that may be large, but also for simple heterogeneous CAS models. Yet, the widespread diffusion of model checking approaches to describe and check complex system properties makes them an attractive approach. One way to scale them up is that of combining them with the fluid flow approximation, obtaining approximate but fast model checking routines.

This idea has been first introduced and worked out in detail in [BH12], where the authors discussed how fluid approximation could be used to efficiently check individual behavioural properties in the context of large population models, an approach they named Fluid Model Checking (FMC). These properties are interesting also for CAS. For instance, one could ask what is the probability for a user to find a bike in a station within a one kilometre radius from her current location in a densely populated city with thousands of shared bikes and hundreds of bike stations. Or one could investigate the chances of a smart boiler to provide hot water when needed. Normally, this kind of properties would have been checked by constructing the global model, tagging an individual agent, and checking properties of this tagged agent. The approach proposed in [BH12], instead, leverages on a corollary of fluid approximation theorems, also known as *fast simulation* [DN08, GG10, BHLM13], which essentially allows us to construct an approximate mean-field model of an individual. This model is obtained by decoupling the single agent from the rest of the system, which is then averaged away and pushed in the background environment, using the fluid flow differential equations. The resulting model is a

[1]Indeed, state space explosion affects numerical and statistical methods in different ways, but both of them have a computational complexity which increases with the number of agents in the system, linearly for statistical model checking, and polynomially or exponentially for numerical methods.

time-inhomogeneous Markov chain, with as many states as those of a single agent, and with rates changing continuously in time, reflecting the solution of the fluid equations.

In particular, in [BH12], the authors considered properties of individual behaviours described by Continuous Stochastic Logic (CSL) [ASB+95, BHHK03], proposing algorithms to solve the model checking problem for CSL for time-inhomogeneous models, which is a challenging problem, especially when nested properties are considered. The core of the proposed method relies on rephrasing the reachability computations that have to be solved to check CSL formulae in terms of the solution of piecewise-continuous Kolmogorov equations, where the discontinuities emerge in time due to the non-constant nature of rates, particularly when one needs to deal with nested formulae. In fact, the time-dependency of rates propagates to the truth value of CSL formulae, which can now change in time, introducing discontinuities in the satisfaction probability. The authors also present decidability results and convergence results for the truth of CSL formulae, which guarantee the asymptotic consistence of the proposed method.

Fast simulation has been employed to analyse specific models in few cases, like in [GG10], in order to study policies for balancing the load between servers in large-scale clusters of heterogeneous processors. In [HSB12, HBC13], instead, the authors propose a method to approximately compute passage time of individual components, captured by probes (deterministic automata) in the context of PEPA models, using fluid approximation techniques.

Markov Population Models. Before proceeding further with the discussion, we briefly introduce here Markov Population Models, which are Continuous Time Markov Chains (CTMC) describing a population of interacting agents. In particular, let $Y_i^{(N)} \in S$ represent the state of agent i, where $S = \{1, 2, \ldots, n\}$ is the state space of each agent. Multiple classes of agents can be represented in this way by suitably partitioning S into subsets, and allowing state changes only within a single class. Here N is the total population size. When dealing with population models, it is customary to assume that single agents in the same internal state cannot be distinguished, hence we can move from the agent representation to the system representation by introducing variables counting how many agents are in each state. With this objective, define[2] $X_j^{(N)} = \sum_{i=1}^{N} \mathbf{1}\{Y_i^{(N)} = j\}$, so that the system can be represented by the vector $\mathbf{X}^{(N)} = (X_1^{(N)}, \ldots, X_n^{(N)})$, whose dimension is independent of N. The domain of each variable $X_j^{(N)}$ is obviously $\{1, \ldots, N\}$. The evolution of the system is described by a set of transition rules at this global level. This simplifies the description of synchronous interactions between agents, still allowing the evolution from the perspective of a single agent to be reconstructed from such a system level dynamics. Each transition $\tau \in \mathcal{T}^{(N)}$ is defined by a *multi-set* of *update rules* R_τ and by a rate function $r_\tau^{(N)}$. We assume that R_τ is independent of N, so that each transition involves a finite and fixed number of individuals. Given a multi-set of update rules R_τ, we can define the *update vector* $\mathbf{v}_\tau$, counting the net change in each population variable due to the occurrence of a τ transition. Hence, each transition changes the state from $\mathbf{X}^{(N)}$ to $\mathbf{X}^{(N)} + \mathbf{v}_\tau$. The rate function $r_\tau^{(N)}(\mathbf{X})$ depends on the current state of the system, and specifies the speed of the corresponding transition. It is required to be *Lipschitz continuous*.

2.2 Objectives

The goals within Work Package 3 of the project, for what concerns fluid approximation techniques for stochastic model checking, are manyfold. First of all, we aim to extend and consolidate the current range of applicability of fluid model checking. This encompasses extending the class of properties that can be checked, from steady state (when possible) to rewards and more complex individual behaviours, for instance combining probes with the CSL logic formalisms.

Another fundamental aspect of QUANTICOL is the link between local and global specifications. This is a fundamental step to move from individual behaviours to collective properties to better

[2]The expression $\mathbf{1}\{x = y\}$ gives $\mathbf{1}$ if $x = y$ and 0 otherwise.

understand and characterise the emergence of collective behaviours. This problem has many flavours: lifting local specifications to global ones, and using fluid approximations to check properties at the global level.

Finally, within the project we aim to investigate how fluid model checking techniques can be applied to CAS, especially to the case studies of the project.

2.3 Achievements in Project Year 1

In the first year of the project, we mainly obtained theoretical achievements. On the one hand, we consolidated the FMC approach by considering the introduction of two further CSL operators. These are the steady state operator (for systems that show steady state behaviour) and the next state operator [BH13b]. On the other hand, we obtained some results in the context of relating local and global properties, developing a new method [BL13] based on a second-order fluid approximation known as linear noise approximation [vK07], which is a functional version of the central limit approximation [Kur81]. We discuss this second approach in more detail in the following.

The Central Limit Way to Model Checking. In [BL13], the authors present a method to lift local specifications to collective ones by means of the central limit theorem. Their starting point is an individual property, specified by a deterministic timed automaton with a single global clock (i.e. with a unique clock variable which is never reset, and hence keeps track of the global passing of time). This specification language captures many interesting linear-time properties, for instance the fact that a smart boiler contains hot water between 6 and 7 pm. Given an individual specification ϕ, then one counts how many agents in the system satisfy ϕ at time t, denoting this number by $X_\phi(t)$, and then asks the following global property:

$$\text{What is the probability that } X_\phi(t) \geq \theta?$$

This class of global properties is highly relevant in CAS models. For instance, in a bike sharing scenario, we can be interested in computing the probability that a large fraction of users find a bike in a station within 1km from their current location.

The method presented in [BL13] allows us to quickly estimate this probability by exploiting the central limit or linear noise approximation. The idea behind this class of fluid methods is to retain some information about the fluctuations of the system. As a general rule of thumb, at a certain time t the fluctuations are expected to be approximately Gaussian, if the populations are sufficiently large. Moreover, the magnitude of such fluctuations is of the order of $\sqrt{N}$, where N is the population size. If we assume that the system fluctuates around its fluid flow limit $\mathbf{x}(t)$, then we can approximate a stochastic population process with N individuals, $\mathbf{X}^{(N)}(t)$ by

$$\mathbf{X}^{(N)}(t) \approx N\mathbf{x}(t) + \sqrt{N}\mathbf{Z}(t),$$

where $\mathbf{Z}(t)$ is a Gaussian stochastic process, i.e. a process whose finite dimensional projection (marginal distributions at any fixed and finite set of times) are Gaussian. This Gaussian Process has zero mean, and a covariance given by the solution of an additional set of $\mathcal{O}(N^2)$ ODEs, which can still be computed efficiently. More details can be found in [BL13, vK07].

The idea to apply the central limit theorem to check global properties of the form introduced above is as follows. First, we combine the automaton-based property specification with the model of an individual agent, by a product construction (taking into account the clock constraints), obtaining a population model with more variables, counting pairs of state-property configurations. Then, the central limit approximation is applied consistently to this new model. It turns out that for a large class of individual properties, we can explicitly introduce a variable $X_\phi(t)$ in the extended model that counts how many individual agents satisfy the local property up to time t. From the Gaussian approximation of $X_\phi(t)$, then one can easily compute the probabilities of interest. In [BL13], the authors discuss preliminary results, which are quite accurate and computationally efficient.

Stability Analysis for Stochastic Process Algebra Fluid flow approximations are not only very valuable for analysis techniques based on model checking, but play an important role also in stability analysis. Stability analysis provides important information about the predictability of dynamic systems and their sensitivity to parameter values. Numeric stability analysis is of particular interest for the analysis of distributed adaptive strategies that are applied in asymmetric situations. Such situations occur naturally in many real-world, natural or designed, systems. For example, if we consider space divided into discrete patches and mobility of objects between such patches, some patches may be more attractive to objects than other patches or maybe more or less easy to reach. An analytical approach to stability analysis in the presence of asymmetry is infeasible in most cases. In [BLM13] a new numeric method, in the form of a tool-chain, is proposed for stability analysis starting from a stochastic process algebra specification of agent coordination in a collective dynamic system. The use of a process algebra greatly facilitates the modelling of variants of a system at a high level. Designers can then focus on the coordination strategies instead of having to manipulate the underlying, possibly large set of non-compositional, ODEs in ad hoc ways. The method is illustrated on a combined numeric and symbolic approach on a number of variants of a model of crowd dynamics that represents various kinds of asymmetry. The automation of the approach is feasible and part of future work.

2.4 Novelty and Future work

The idea of using an approach based on functional central limit results to efficiently approximate some classes of stochastic model checking problems is novel, and has a lot of potential. This method can be extended in many directions: enlarging the set of local properties that can be considered, enriching the ways local properties are lifted to global ones, considering higher order corrections to improve accuracy. Similarly, ideas based on the application of the central limit theorem could be used to check global properties, where atomic propositions are inequalities on system variables, which can be reduced to reachability problems in the population state space. Additional work is also needed to consolidate the original fluid model checking methodology, looking at individual properties. More specifically, we plan to extend the class of properties that can be checked, and connect the method with languages developed in the context of the QUANTICOL project, see Deliverable 4.1. An additional topic we plan to investigate is the extension to the spatial setting (see Deliverable 2.1) and to spatio-temporal properties (see Section 5) of the individual and local-to-global fluid model checking approaches. Finally, we plan to consider the applicability of all these techniques to CAS and to the case studies of the project, in particular in the context of smart transportation. This will possibly suggest new theoretical directions to investigate, and in any case it requires a working implementation of the methods, which is a top priority task.

3 Probabilistic logic and model checking techniques that exploit mean-field techniques

The FMC approach presented in Section 2 is based on a global model checking algorithm and on fluid approximation in continuous time. In this section we provide a brief account of results obtained during the first year of the project in the area of probabilistic logic (PCTL) on-the-fly model checking techniques that exploit mean-field approximation results in discrete time. All technical details can be found in [LLM13a, LLM13b, LLM13c]. Knowledge of PCTL [HJ94] is assumed in the sequel. Furthermore, by bounded PCTL we mean the logic composed of all modalities of PCTL except Unbounded Until.

3.1 State of the art/base line before start of project

An on-the-fly model checking approach by itself does not solve the challenging scalability problems that arise in truly large parallel systems, such as collective adaptive systems. To address this type of

scalability challenges in probabilistic model checking, recently, several approaches have been proposed. In [HLMP04, GHLP06] approximate probabilistic model checking is introduced. This is a form of statistical model checking that consists of the generation of random executions of an *a priori* established maximal length. On each execution the property of interest is checked and statistics are performed over the outcomes. The number of executions required for a reliable result depends on the maximal error-margin of interest. The approach relies on the analysis of individual execution traces rather than a full state space exploration and is therefore memory-efficient. However, the number of execution traces that may be required to reach a desired accuracy may be large and therefore time-consuming. The approach works for general models, i.e., not necessarily populations of similar objects, but is not independent of the number of objects involved.

To analyse properties of large scale mobile communication networks mean-field approximations in discrete time have been used e.g., in Bakhshi et al. [BEE$^+$10]. In that work an automated method is proposed and applied to the analysis of dynamic gossip networks. A general convergence result to a deterministic difference equation is used, similar to that in [LMM07], but not its extension to analyse individual behaviour in the context of a large population, nor its exploitation in model checking algorithms.

In Chaintreau et al. [CLR09], mean-field convergence in *continuous time* is used to analyse the distribution of the age of information that objects possess when using a mix of gossip and broadcast for information distribution in situations where objects are not homogeneously distributed in space. An overview of mean-field interaction models for computer and communication systems by Benaïm et al. can be found in [BL08].

Preliminary ideas on the exploitation of mean-field convergence in *continuous time* for model checking mean-field models, and in particular for an extension of the logic CSL, were informally sketched in a presentation at QAPL 2012 [KRd12], but no model checking algorithms were presented. In [BH12] Bortolussi and Hillston proposed a global fluid model checking algorithm for the verification of CSL properties of a selection of individuals in a population (see also Sect. 2 of this Deliverable). Follow-up work on [KRd12] can be found in [KRd13], which relies also on [BH12].

Earlier work by Stefanek et al. on the use of mean-field convergence in *continuous time* for an extension of the Performance Evaluation Process Algebra to deal with groups of objects (grouped PEPA) has investigated the quality of the convergence results when the related differential equations are derived directly from the process algebraic model [SHB10].

3.2 Objectives

One of the main objectives for the first year of the project was to develop novel, scalable formal analysis techniques and the underlying theories to support the design of CAS consisting of very large numbers of autonomous heterogeneous components. We were aiming at novel forms of scalable model checking approaches that exploit mean-field and fluid approximations. In this section we present our achievements concerning the development of a novel model checking procedure, based on an original combination of local, on-the-fly model checking techniques and fast mean-field approximation for discrete time, time-synchronous, probabilistic population models. A further aim was to develop this approach first for well-established probabilistic temporal logics such as PCTL, addressing properties of selected individual objects in the overall system, and to start investigating the extension of such a logic with global properties, addressing the average overall behaviour of the system, and mixed local and global properties. Scalability of the approach is intended as scalability in terms of the size of the population of objects. In other words, the approach should be insensitive to the size of the population comprising the system.

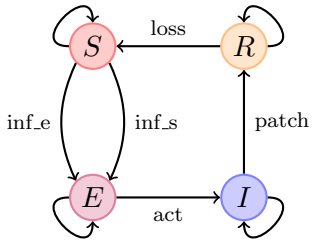


Figure 1: Epidemic model object.

3.3 Achievements in Project Year 1

During the first year of the project, the foundations of an on-the-fly fast mean-field model checking procedure have been developed. The procedure is used for verifying bounded PCTL formulas expressing properties of an object (or a limited number of objects), in the context of systems consisting of a large number of independent but interacting objects. The approach is based on the mean-field approximation and fast-simulation results of [LMM07]. Following the approach in [LMM07] we consider a model for interacting objects, where the evolution of each object is given by a finite state discrete time Markov chain (DTMC). A simple language for system specifications is provided. The behaviour of a generic object of a system is defined by means of two sets of defining equations. In Fig. 1 an object is shown for a computer epidemic model (described in [BHLM13] in detail). The object represents a node of a computer network which can acquire infection from two sources, i.e. by the activity of a worm of an infected node (`inf_s`) or by an external source (`inf_e`). Once a computer is infected, the worm remains latent for a while, and then activates (`act`). When the worm is active, it tries to propagate over the network by sending messages to other nodes. After some time, an infected computer can be patched (`patch`), so that the infection is recovered. New versions of the worm can appear; for this reason, recovered computers can become susceptible to infection again, after a while (`loss`). An example of *object specification*, in our simple language, of the object of Fig. 1 is given in Fig.2. It consists of a set of defining equations for the states of the object (a), the meaning of which should be obvious, and a set of equations for defining the probabilities with which the object may execute its actions (b). The probabilities are defined by means of numerical expressions[3] which may refer also to the *fraction* of objects which are currently in a specific state, in the system the object is an element of. In the example, $\mathsf{frc}\,\mathtt{I}$ denotes the fraction of objects in state `I`, in the system at hand.

```
S := inf_e.E + inf_s.E
E := act.I
I := patch.R
R := loss.S
```

(a) State definitions

$$\begin{aligned}
&\texttt{inf_e} :: \alpha_e;\\
&\texttt{inf_s} :: \alpha_i * (\mathsf{frc}\, I);\\
&\texttt{act} :: \alpha_a;\\
&\texttt{patch} :: \alpha_r;\\
&\texttt{loss} :: \alpha_s;
\end{aligned}$$

(b) Action probability definitions

Figure 2: Object specification

A *system specification* is composed by an N-tuple of instances of objects, together with the specification of the initial state for each instance[4]. N is a system specification parameter, denoting the *size*

[3] In the example we use symbolic parameters α_s for constants.

[4] More precisely, a system specification may consist of one or more object specifications and a total of N instantiations of it. For the sake of simplicity, in this document we consider the case with only one object specification (and N instantiations, each in a possibly different initial state).

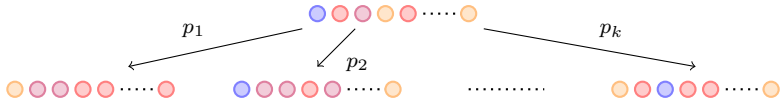


Figure 3: A fragment of the $\mathcal{D}^{(N)}(t)$ transition relation.

of the system. A system *global state* $\mathbf{C}^{(N)} = \langle c_1, \ldots, c_N \rangle$ is the N-tuple of the current local states of its object instances; each object can be in one of its local states at any point in time and all objects proceed in discrete time and in a clock-synchronous fashion. The behaviour of each object is characterised by a transition matrix $\mathbf{K}^{(N)}$ which is mechanically derived from the specification and which defines the state transitions of the object and their probabilities. Thus the transition matrix of each object may depend on the distribution of states of *all* objects in the system. More specifically, $\mathbf{K}^{(N)}$ is a function $\mathbf{K}^{(N)}(\mathbf{m})$ of the *occupancy measure* vector $\mathbf{m}$ of the current global state $\mathbf{C}^{(N)}$. Under the assumption that the set of object states is a total order of cardinality S, the occupancy measure vector $\mathbf{m} = \langle m_1, \ldots, m_S \rangle$ is an element of $\mathcal{U}_S = \{\langle r_1 + \ldots + r_S \rangle \in [0,1]^S \mid r_1 + \ldots + r_S = 1\}$ and $m_i = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}_{\{c_n = c_i\}}$ denotes the fraction of local states c_i in the global state $\mathbf{C}^{(N)} = \langle c_1, \ldots, c_N \rangle$.

For model checking purposes different *(state) labels* can be associated with (different) states of an object in its specification; the global states of a system will be automatically labelled with the label of the first component of the N-tuple.[5] Furthermore, *global* system properties can be expressed using atomic propositions p defined by equations of the form $p = F(E_1, \ldots, E_h)\ \texttt{relop}\ r$, where F is a continuous function from $[0,1]^h$ to $\mathbb{R}$, each E_j is of the form $\mathsf{frc}\, c$, for object local state c, and $r \in \mathbb{R}$ (see [LLM13a, LLM13b, LLM13c] for details). In this way, the *state labelled* DTMC $\mathcal{D}^{(N)}(t)$ of the system is defined using the transition matrix generated from the object specifications and the global initial state. Fig. 3 gives an idea of a fragment of the one step probability matrix of $\mathcal{D}^{(N)}(t)$, for the computer epidemic model, with reference to Fig. 1.

The model checking algorithm we developed is *parametric with respect to the semantic interpretation* of the language. In fact, the on-the-fly model checking algorithm and its implementation are independent from the specific modelling language because they are defined on an abstract interpreter. In particular we have been carrying out experiments with two different instantiations. The first instantiation is on *exact probabilistic semantics* of the language sketched above in order to check $\mathcal{D}^{(N)}(t)$ against (full) PTCL formulas. The second instantiation is on *approximate mean-field semantics* of the language that is addressed in the following. When N is very large, exact model checking, even on-the-fly, is not feasible. On the other hand, for N large, the overall behaviour of the system in terms of its occupancy measure can be approximated by the (deterministic) solution of a *difference equation* which is called the 'mean-field'. This convergence result has been extended in [LMM07] to obtain a 'fast' way to stochastically simulate the evolution of a selected, limited number of specific objects in the context of the overall behaviour of the population.

We showed that the deterministic iterative procedure of [LMM07], to compute the average *overall* behaviour of the system and the behaviour of *individual* objects in the *context* of the system, combines well with an *on-the-fly* probabilistic model checking procedure for the verification of bounded PCTL formulas addressing properties of selected objects of interest. Note that the transition probabilities of these selected objects at time t may depend on the occupancy measure of the system at t and therefore also the truth-values of the formulas may vary with time. Furthermore, the set of atomic formulas can be extended along the lines proposed in [KRd12, KRd13] in order to express some properties of the system *global* state as well. We have shown a simple example of that in [LLM13a].

Our procedure first of all derives a simplified view of $\mathcal{D}^{(N)}(t)$ by abstracting from the details of all the N object instantiations except from those of the first object and the global occupancy

[5] Although, in principle, the model checking technique can be used for formulas describing local properties of any limited set of k objects, here, for simplicity, we consider only the case $k = 1$.

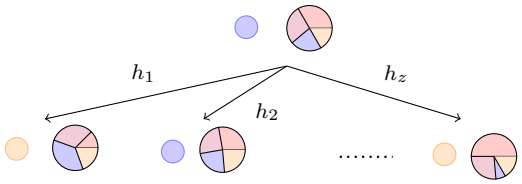


Figure 4: A fragment of the $\mathcal{HD}^{(N)}(t)$ transition relation.

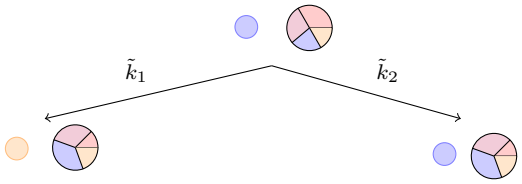


Figure 5: A fragment of the $\mathcal{HD}(t)$ transition relation.

measure. In practice we define a mapping $\mathcal{H}^{(N)}$ on the states of $\mathcal{D}^{(N)}(t)$ such that, for each global state $\mathbf{C}^{(N)} = \langle c_1, \ldots, c_N \rangle$, $\mathcal{H}^{(N)}(\mathbf{C}^{(N)}) = \langle c_1, \mathbf{m} \rangle$ where $\mathbf{m}$ is the occupancy measure vector of $\mathbf{C}^{(N)}$. It is easy to see that mapping $\mathcal{H}^{(N)}$ induces another state labelled DTMC, $\mathcal{H}^{(N)}(\mathcal{D}^{(N)}(t))$, which we denote by $\mathcal{HD}^{(N)}(t)$. Fig. 4 gives an idea of $\mathcal{HD}^{(N)}(t)$ for $\mathcal{D}^{(N)}(t)$ as in Fig. 3; occupancy measure vectors are represented as "pie-charts".

Mapping $\mathcal{H}^{(N)}$ preserves logic satisfaction[6], that is $\mathbf{C}^{(N)} \models_{\mathcal{D}^{(N)}} \Phi$ iff $\mathcal{H}(\mathbf{C}) \models_{\mathcal{HD}^{(N)}} \Phi$, for all $N > 0$, states $\mathbf{C}$ of $\mathcal{D}^{(N)}(t)$ and bounded PCTL formulas Φ.

We now consider the stochastic process $\mathcal{HD}(t)$ defined below, for c_0, c, c' object local states, $\boldsymbol{\mu}_0, \mathbf{m}, \mathbf{m}' \in \mathcal{U}_S$, where $\mathcal{U}_S$ is the unit simplex of dimension S, $\boldsymbol{\delta}_{\langle c, \mathbf{m} \rangle}$ the Dirac distribution with all probability mass in $\langle c, \mathbf{m} \rangle$, and function $\mathbf{K}(\mathbf{m})_{c,c'}$, continuous in $\mathbf{m}$:

$$\begin{aligned} &\Pr\{\mathcal{HD}(0) = \langle c, \mathbf{m} \rangle\} = \boldsymbol{\delta}_{\langle c_0, \boldsymbol{\mu}_0 \rangle}(\langle c, \mathbf{m} \rangle), \\ &\Pr\{\mathcal{HD}(t+1) = \langle c', \mathbf{m}' \rangle \,|\, \mathcal{HD}(t) = \langle c, \mathbf{m} \rangle\} = \begin{cases} \mathbf{K}(\mathbf{m})_{c,c'}, & \text{if } \mathbf{m}' = \mathbf{m} \cdot \mathbf{K}(\mathbf{m}) \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{1}$$

Labelling each state $\langle c, \mathbf{m} \rangle$ with the label of c, we note that also $\mathcal{HD}(t)$ is a state labelled DTMC. In the sequel, we consider sequences of system specifications for $N > N_0$, for some N_0; in particular, we will focus on the related sequences $\left(\mathcal{D}^{(N)}(t)\right)_{N \geq N_0}$ of the associated DTMCs and $\left(\mathcal{HD}^{(N)}(t)\right)_{N \geq N_0}$ of their abstractions; for each N, we let $\mathbf{C}_0{}^{(N)}$ denote the initial state of $\mathcal{D}^{(N)}(t)$. The following is a corollary of Theorem 4.1 (Deterministic approximation) and Theorem 5.1 (Fast simulation) of Le Boudec *et al.* [LMM07], when considering sequences $\left(\mathcal{HD}^{(N)}(t)\right)_{N \geq N_0}$ as above: assume that for all object local states c, c', there exists function $\mathbf{K}_{c,c'}$, continuous in $\mathbf{m}$, such that, for $N \to \infty$, $\mathbf{K}^{(N)}(\mathbf{m})_{c,c'}$ converges uniformly in $\mathbf{m}$ to $\mathbf{K}(\mathbf{m})_{c,c'}$; assume, furthermore, that there exists $\boldsymbol{\mu}_0 \in \mathcal{U}_S$ such that the occupancy measure vector of $\mathbf{C}_0^{(N)}$ converges almost surely to $\boldsymbol{\mu}_0$; define function $\boldsymbol{\mu}(t)$ of t as follows: $\boldsymbol{\mu}(0) = \boldsymbol{\mu}_0$ and $\boldsymbol{\mu}(t+1) = \boldsymbol{\mu}(t) \cdot \mathbf{K}(\boldsymbol{\mu}(t))$; then, for any fixed $\hat{t}$, almost surely, $\lim_{N \to \infty} \mathcal{HD}^{(N)}(\hat{t}) = \mathcal{HD}(\hat{t})$.

Process $\mathcal{HD}(t)$ is often called the *limit approximation* of $\mathcal{HD}^{(N)}(t)$. Fig. 5 sketches a fragment of the transition relation of $\mathcal{HD}(t)$, with reference to $\mathcal{HD}^{(N)}(t)$ of Fig 4. Before stating our main result, which uses the above corollary, we need a little bit of further notation. A path σ over DTMC $\mathcal{M}$ is a non-empty sequence of states $s_0, s_1, \cdots$ with $\mathbf{P}_{s_i, s_{i+1}} > 0$ for all $i \geq 0$, where $\mathbf{P}$ is the transition probability matrix of $\mathcal{M}$. We let $Paths_{\mathcal{M}}(s)$ denote the set of all infinite paths over $\mathcal{M}$ starting from state s. A formula Φ is *safe* for a model $\mathcal{M}$ iff for all sub-formulas Φ' of Φ and states s of $\mathcal{M}$, if Φ' is

[6] For state s of model $\mathcal{M}$ and formula Φ, the notation $s \models_{\mathcal{M}} \Phi$ expresses the fact that s satisfies Φ in model $\mathcal{M}$.

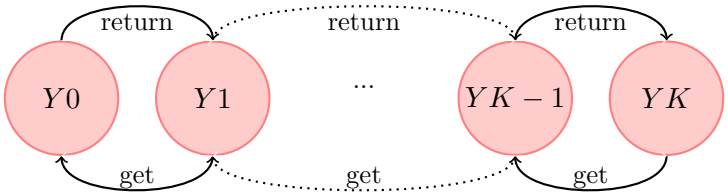


Figure 6: Bike station.

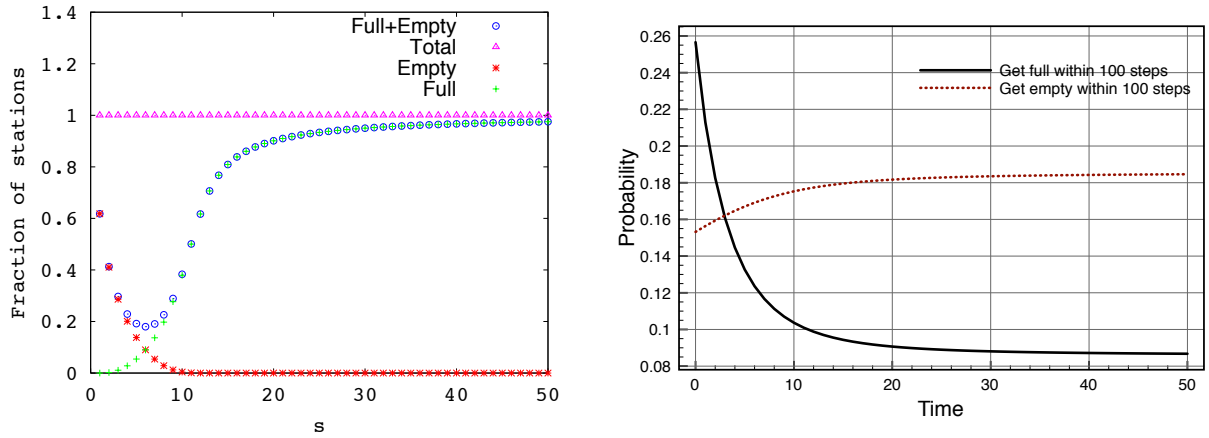
of the form $\mathcal{P}_{\bowtie p}\varphi$ then $\Pr\{\eta \in Paths_{\mathcal{M}}(s) \mid \eta \models_{\mathcal{M}} \varphi\} \neq p$, where $\bowtie \in \{\geq, >, \leq, <\}$. Our main result can be summarised as follows: let $\mathbf{C}_0^{(N)} = \langle c_1, \ldots, c_N \rangle$ be the initial global state of $\mathcal{D}^{(N)}$ and Φ a safe bounded PCTL formula. Then, under the assumptions of Theorem 4.1 of [LMM07], for any fixed $\hat{t}$, almost surely, for N large enough, formula satisfaction is preserved in the limit approximation, i.e. $\mathbf{C}_0^{(N)} \models_{\mathcal{D}^{(N)}} \Phi$ iff $\langle c_1, \boldsymbol{\mu}(\hat{t}) \rangle \models_{\mathcal{HD}} \Phi$.

Process $\mathcal{HD}^{(N)}$ is used as an intermediate representation of $\mathcal{D}^{(N)}$ in the formal proof of the above result. It is worth noting that, although safety for all sub-formulas of Φ is required, in practice it is sufficient to check for safety only those sub-formulas which are encountered during the execution of the on-the-fly model checking procedure.

The asymptotic correctness of the model checking procedure has been proven and a prototype implementation of the model checker, FlyFast, has been applied to the computer epidemic model [BHLM13], to a bike-sharing model inspired by the model by Fricker and Gast [FG13] and to a classical predator-prey model by Lotka and Volterra [Lot24, Vol26]. In the following we briefly address the application of the FlyFast model checker to the bike-sharing case study. The original bike-sharing model by Fricker and Gast is based on homogeneous space where all locations are equally accessible to the users and is a continuous time model. We have used a discrete time variant of this model consisting of N stations, each with a capacity of K parking slots. The number of bikes in the system is constant with s bikes per station on average, amounting to a total of $s \cdot N$ bikes. In every time step each station has the same probability that a user retrieves a bike. The probability that a bike is returned to a station depends on the number of bikes that are in circulation (i.e. not parked). Fig. 6 shows a graphical representation of the model of a single bike station with K parking slots.

One of the interesting aspects of the system is the probability that stations are empty or full, since these situations would generate dissatisfaction among users and would discourage them from using the system in the future. We have used FlyFast to analyse both global aspects of the system as well as properties of an individual station. An example of the first is to find the optimal number of bikes in the system, assuming certain values for the probability with which bikes are requested. The result is shown in Fig. 7(a), where the fraction of problematic stations are shown at time step 100, a point in time in which the system is nearly stable, for an average number of bikes per station s ranging from 1 to 50. These results are compatible with those in [FG13]. An example of a property of an individual station, operating in the context of the global system, is the question how likely it is that a station gets full or empty within 100 steps. This property can be formalised in PCTL as $\mathcal{P}_{=?}(\ \mathrm{tt}\ \mathcal{U}^{\leq 100}\ Y10\ )$ and similarly for getting empty, for stations with 10 parking slots ($K = 10$). Assuming that initially all stations are empty, the black curve in Fig. 7(b) shows the probability that the station gets full within 100 steps. Note that this probability depends on the time (in steps) at which the property is evaluated. Similarly, the figure shows the results for the probability that the station under study gets empty when it is initially full. The interested reader is referred to [LLM13c] for further details and examples, including the analysis of the effect of user incentives, again inspired by the work by Fricker and Gast [FG13] but now cast in a mean-field fast on-the-fly PCTL model checking setting.

This model checking technique is ultimately based on the mean-field approximation in discrete time of the limit behaviour of the underlying stochastic population Markov process by a discrete-



(a) Fraction of problematic stations

(b) Individual station getting empty (full, resp.)

Figure 7: Bike sharing: FlyFast model checking results

time dynamical system. An important question is how well such an approximation works, or more precisely, what guaranteed estimates can be obtained of the deviation of the stochastic process from its deterministic limit. This question is addressed in [BH13a] for the transient dynamics and the steady state of certain classes of discrete-time population Markov processes. The proposed method combines stochastic bounds in terms of martingale inequalities and Chernoff inequalities, with control-theoretic methods to study the stability of a system perturbed by non-deterministic noise terms, and with algorithms to over-approximate the set of reachable states. The insight behind this method is that, by abstracting stochastic noise in a non-deterministic fashion, refined control theoretic tools can be used that can take into account the fine-grained nature of the phase space of the mean field limit. The latter is largely ignored by the previously known approaches to compute error bounds and they therefore result in very loose bounds, restricted to transient behaviour, that expand exponentially with the time interval considered. The new method instead, allows us to obtain for the first time general bounds for steady-state convergence, and, additionally transient error bounds which do not explode exponentially. The interested reader is referred to [BH13a] for further details.

3.4 Novelty and Future work

Although the work presented in Sect. 2 has some basic ideas in common with the work described in this section, there are some fundamental differences: the former exploits mean-field convergence and fast simulation in a continuous time setting [DN08, GG10], while the latter does this in a *discrete time* setting. The former is based on an interleaving model of computation, whereas the latter on a *clock-synchronous* one; furthermore, the former uses a *global* model checking approach, whereas the latter an *on-the-fly* approach. To the best of our knowledge, FlyFast [LLM13a] is the first implementation of an *on-the-fly fast mean-field* model checker for (bounded) PCTL on *discrete time, probabilistic, time-synchronous* models for one or several objects living in an environment with a very large population, addressing both local – object-level – properties as well as a limited class of global – system-level – ones. Furthermore, in [BH13a] a new method has been developed to obtain more precise error bounds, both for steady state convergence and for transient analysis of discrete-time population Markov models.

The work that has been carried out during the first year was mainly foundational. We plan to stabilise, complete and finalise the results in future work. Aspects of interest, which we plan to investigate include the incorporation of notions of *space* (see Deliverable 2.1) both in the modelling language and in the logic (see Section 5), in order to also take physical space into consideration when analysing collective adaptive systems in a discrete time and space setting. Moreover, many realistic

system models would include a hierarchical system description leading to models of systems-of-systems. Both extensions are likely to lead to larger models for which suitable model-reduction techniques would be extremely important to obtain scalable verification beyond population size.

Another aspect which requires further study is the definition of a higher level compositional modelling language for CAS (see Deliverable 4.1) that extends the simple language used in the first phase of the project. Further extensions could include costs and rewards and error estimates. Also a more detailed study of the relationship between Global Fluid Model Checking and On-the-fly Mean-Field Model Checking could be of interest.

4 Statistical model checking approaches

4.1 State of the art/base line before start of project

Model checking logical properties against stochastic models is in practice limited by the state space explosion, which hinders the ability of numerical model checking approaches, like PRISM [KNP04] to tackle large models, such as those of heterogeneous CAS systems. One line of attack to this problem, which received a lot of attention in the last ten years, is that of statistical model checking (SMC) [YS02, YS06, YKNP04, JCL$^+$09, ZPC10]. The idea behind SMC is simple: instead of considering the whole state space, or portions of it, just look at a number of simulated trajectories of the system and extract from them statistical information about the truth of logical statements. SMC works best for linear time properties, for which it can be used to estimate their satisfaction probability, or to check if such a probability is above or below a given threshold. In particular, it is the only current feasible approach for linear time properties with metric time bounds, such as Metric (interval) Temporal Logic (MiTL) [Hen98] and its real-time specialisation Signal Temporal Logic (STL) [MN04].

More precisely, SMC works as follows. Given a stochastic process in continuous time (not necessarily a CTMC, this works also for stochastic hybrid systems) and a STL formula ϕ, the model is simulated N times, checking for each trajectory so generated if ϕ is true or false. This procedure, in fact, produces independent samples from a Bernoulli random variable, with probability equal to the satisfaction probability of ϕ. Given those samples, we can use standard statistical tools to either estimate such a probability or to test if it is greater or smaller than a given threshold. This can be done either in the frequentist statistics setting, like in [YS02, YS06, YKNP04], which uses Chernoff-like bounds [Bil95] and Wald sequential sampling test [Wal45], or in a Bayesian setting, as in [JCL$^+$09, ZPC10], which exploits properties of the Beta distributions and Bayesian model selection. An important aspect of those methods is that they also provide error estimates, for a given confidence, of the estimation/ testing procedure.

More refined approaches to SMC look at problems like the estimation of very small or very large probabilities, using importance sampling techniques [JLS12], or they integrate SMC into optimisation routines to search for model parameters that satisfy with high probability the properties of interest. In [DG08], for instance, the authors use an evolutionary computation search strategy. In a numerical MC context, the problem of parameter uncertainty has been considered in [KKLW07], where authors essentially treat uncertain CTMC as Continuous-Time Markov Decision Processes, and in [BD13], where a method based on iterative splitting of intervals for CSL formulae is developed. However, both methods are highly computationally intensive and do not scale to the systems of interest in QUANTICOL.

A different approach towards system design and treatment of parameter uncertainty, which however was never considered in the context of stochastic models, but rather on continuous deterministic ones, is based on the idea of associating with a formula a measure of robustness, i.e. a real number which is positive when the formula is true, negative when the formula is false, and its magnitude measures how robust is such a truth value. We recall here the work of Fages [RBFS11], that of Pappas and Faneikos [FP09], and the robust semantics of STL by Maler and Donzé [DFM13, DM10], all differing in some details about the definition of the robustness score.

4.2 Objectives

In the context of QUANTICOL, we are looking for efficient routines that can deal with uncertainty in model parameters in a statistically sound way. SMC is a natural candidate in this respect, as it can scale up, in principle, to very large systems, given efficient simulation algorithms. Furthermore, it is trivially parallelisable, hence amenable to High Performance Computing techniques. However, running many simulations for many points in the parameter space could be a daunting task also for modern parallel computation facilities. Hence, there is the need to investigate new methods to deal with parameter uncertainty in a more computationally feasible way.

We also need to investigate efficient system design algorithms, as this will likely be a key computational routine to control CAS models. SMC is likely to play a strategic role also in this respect.

4.3 Achievements in Project Year 1

In the first year of the project we investigated both problems, making some relevant advancements. We will first sketch the ideas for the system design problem, which are taken from the paper [BBNS13], where we refer for more details, and then present a new statistical approach to deal with parameter uncertainty. We conclude discussing a novel SMC tool developed in the context of the project.

System Design. The idea at the base of our approach is to export the definition of STL robust semantics [DFM13, DM10] to the stochastic setting of Markov population models, and suitably define the system design problem within such a context. This results in an optimisation problem, which we solved with state-of-the-art optimisation methods borrowed from the machine learning literature.

Let us sketch the above mentioned approach in more detail. First of all, we recall the definition of Signal Temporal Logic, a variant of Metric Temporal Logic (linear time logic on continuous time, with time bounds on the temporal operators) which is suitable for systems whose variables take real values. The syntax of STL is as follows:

$$\phi ::= \mu \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \phi_1 \mathbf{U}^{[T_1,T_2]} \phi_2,$$

where $\mu = \mu(\mathbf{X}) \equiv f(\mathbf{X}) \geq 0$ is an atomic predicate, i.e. a (non-)linear inequality on system variables, and $\mathbf{U}^{[T_1,T_2]}$ is the standard until operator, with the additional constraint that the formula ϕ_2 has to hold between times T_1 and T_2. Eventually ($\mathbf{F}^{[T_1,T_2]}\phi \equiv \mathtt{true}\mathbf{U}^{[T_1,T_2]}\phi$) and globally ($\mathbf{G}^{[T_1,T_2]}\phi \equiv \neg\mathbf{F}^{[T_1,T_2]}\neg\phi$) modalities are defined as usual. The boolean semantics of STL [MN04] is given in terms of boolean signals, which are (right-continuous with left limits) functions of time into boolean values, with a finite number of jumps in any finite time interval. Then, modal and boolean operators are interpreted as transformers of such boolean signals. For instance, the conjunction of two signals is just the point-wise boolean conjunction, for each time instant t, i.e. the point-wise minimum of the two signals. The semantics of modal operators is given in terms of the classical semantics, by replacing universal and existential quantification with a conjunction/ disjunction on a time interval, which itself can be expressed in the usual way in terms of min and max functions. The quantitative semantics of STL [DFM13, DM10], instead, is obtained from the boolean one in terms of max and min by changing the way atomic predicates $\mu = \mu(\mathbf{X}) \equiv f(\mathbf{X}) \geq 0$ are interpreted. In practice, the truth value is replaced by the real number $f(\mathbf{X})$. Note that the predicate is true only when this number is positive, and false otherwise. The use of minimum and maximum preserves this property, so that, given a trajectory, if we obtain a positive number, we can claim that the formula is true. Furthermore, the absolute value of the robustness score of a formula gives a measure of how much the formula is true or false. See [DFM13, DM10] for further details.

In [BBNS13], we lifted this notion of quantitative semantics in a stochastic setting. The idea is simple: for a Markov population model, i.e. a stochastic model in which we describe, by a set of variables how a population of agents interacts, see Section 2, we can consider its induced probability distribution on the set of all possible trajectories and then lift it to a real-valued random variable,

by computing for each trajectory its quantitative satisfaction score (this is the so-called push-out measure). In essence, we obtain a distribution of robustness measures, which can give a lot of information about the system under consideration (for instance, it gives a clear account of bistability, see [BBNS13]). This distribution is essentially impossible to reconstruct analytically, hence we resort to standard statistical tools. More precisely, from a set of simulations, we reconstructed an estimate of the probability density, which is visually captured by the empirical histogram.

Although such a probability distribution is highly informative, it is not suitable to be used for the system design problem, where we need a real-valued function to optimise. Informally, the system design problem we wish to solve is as follows: given a STL formula ϕ, and a Markov population model $\mathcal{M}_\theta$, depending on a set of parameters θ, find a parameter combination θ^* such that $\mathcal{M}_{\theta^*}$ satisfies *robustly* and with *high probability* the formula ϕ. Hence, we need to find a parameter configuration such that ϕ has both a high satisfaction probability and a large average robustness degree (conditional on ϕ being true). To tackle this problem, we analysed a few examples empirically, and found that the unconditional average robustness score is highly correlated with the satisfaction probability, so that maximising the former, we maximise also the latter. This gives the real quantity to optimise for the system design problem. Furthermore, by running many simulations and computing the robustness score for each of them, we can obtain the average robustness with a given error bound, for a fixed confidence level.

However, each function evaluation carried out in this way is expensive and gives a noisy answer. Furthermore, we have no knowledge of the gradient of the function, which is an important ingredient of many optimisation methods. To solve this optimisation problem efficiently, we relied on a Bayesian optimisation strategy originally developed in the context of reinforcement learning, called Gaussian Processes - Upper Confidence Bound optimisation (GP-UCB) [SKKS12]. The idea behind this algorithm is to evaluate the true function in a few points, and then use the information so-obtained to emulate the true function. This is done using statistical tools (Gaussian Process regression [RW06]) that take noise into account in a principled way and return a function for which we have an analytic expression and an estimate of the uncertainty in the reconstruction. The algorithm, however, does not optimise the emulated function (the mean function), but rather an upper quantile of the predictive distribution (for instance, the mean plus 2 standard deviations). If a new maximum is found, the true function is sampled for this point, and such information is used to perfect the emulation in the next stage of the algorithm. The crucial insight is that the emulated function is easy to optimise (compared to the true function). Furthermore, the maxima of the upper quantile can be either from regions in which the true function has a high value (high mean) or in which the uncertainty is high (high standard deviation). In the latter case, a new sample will provide information on an unexplored region, obtaining in this way an automatic balance between exploration and exploitation in the search.

The preliminary results obtained in [BBNS13] are encouraging, showing that the proposed combination of robustness score and optimisation procedure is able to identify values of parameters in which a target property is satisfied robustly with very high probability.

Statistical Model Checking of Uncertain Markov Models. Another problem we tackled during this first year is related to QUANTICOL Work Package 1 and in particular to uncertainty management in CAS modelling. We considered models in which parameters are uncertain, i.e. they are not known precisely but are assumed to be within a given range of possible values. If we have a temporal property, for instance expressed by a MiTL formula, and we want to model check it against such a model, we have to keep uncertainty into account: we cannot identify a precise value for its satisfaction probability, but rather this value will be a function of model parameters.

Previous approaches to model check this class of systems aimed either at finding upper and lower bounds for the satisfaction probability in regions of the parameter space [KKLW07, BD13] or to find closed form expressions for the dependency of the satisfaction probability on model parameters (the so-called parametric model checking [HHZ11]). Both kinds of approach, however, suffer severely

from state space explosion. While classical statistical model checking offers a viable alternative for large models with fixed parameter values to numerical model checking algorithms, applying it to uncertain models is not feasible: even sampling the parametric function in a regular grid is a very costly operation.

Our proposed approach, instead, takes a different perspective, based on recent advances in machine learning and pattern recognition. Consider again the problem we want to solve: we have to find a function of the parameters, which we cannot compute analytically, but only estimate in a finite set of points, with a certain imprecision (SMC produces a noisy output). The idea is to leverage the smoothness of the satisfaction probability function: the knowledge of the satisfaction probability in one point also gives us information about the satisfaction probability for nearby values of parameters. Hence, we can try to exploit the information contained in a finite set of points to estimate the true function from them, using statistical tools that can deal consistently with noise and uncertainty in estimates. More precisely, we tackled this problem in a non-parametric Bayesian setting using Gaussian Processes (GP), which are distributions on function spaces, see [RW06]. Roughly speaking, the idea is to perform a regression and obtain a predictive function (the mean of the GP) together with information about uncertainty in the reconstruction (which is not the uncertainty in the model parameters, but the intrinsic statistical imprecision in the reconstruction of the satisfaction probability value for each parameter value). As the function we have to learn is constrained to be in $[0, 1]$, assigning a probability value to each point, and GP-regression methods work with real valued function, it is natural to map probability values into real numbers using a standard transformation, like the inverse logit or the inverse probit [RW06]. However, we can do better than this using a hierarchical statistical model. We assume the following generative model for the observed data: we start from a GP process, shrunk in $[0, 1]$ by a probit transformation, that gives the satisfaction probability of a Bernoulli random variable, and then sample from this random variable at different values of the parameter space. Hence, we assume that input data are truth values of single simulation runs at different parameter values, so that our data is binary. This makes the model checking problem closer to a classification problem, rather then a regression problem. This intuition can be formalised precisely, and the SMC for uncertain models can be rephrased as a GP-classification problem, under a probit transformation, which can be solved using efficient approximate methods like Expectation-Propagation [RW06]. We adapted these methods to the specific features of the logic-based scenario, and ran some preliminary tests in [BS14], with encouraging results. In particular, we found that we can reconstruct a good approximation of the true satisfaction probability function (depending upon a single parameter) with a limited number of runs, in the order of a few hundreds, hence with a very small computational burden. Furthermore, the use of GP-based statistical tools gives not only the estimate of the satisfaction probability function, but also error bounds on the estimate, for a given confidence level.

Hypothesis Tests for Statistical Model Checking. Hypothesis tests commonly used for SMC aim at verifying whether the probability that a linear time property holds is higher (or lower) than a given boundary value. One way to do this is to use a 'pure' hypothesis test such as the one proposed by Wald [Wal45], which results in a 'yes' (the probability is indeed higher than the boundary) or a 'no'. These tests depend on a so-called 'indifference region', which means that the true probability must be sufficiently far away from the boundary value. Another way is to construct a confidence interval and check whether the boundary value is inside the interval; if so, a 'don't know' answer is returned, otherwise 'yes' or 'no' is returned, depending on how the boundary compares to the interval.

In [RdBSH14], we have introduced a common framework for comparing tests of both types, and have proposed a way of parameterising the confidence interval based tests such that the probability of getting a 'don't know' answer has an upper bound. Furthermore, we discuss two tests that are fundamentally safer (and, unfortunately, slower) than the other tests in the sense that they never return a 'don't know' but also do not need an indifference region.

The MultiVeStA tool. A further contribution in the context of Statistical Model Checking consists of the development of MultiVeStA[7] [SV13], a Java tool extending PVeStA [AM11] and VeStA [AMS06, SVA05], allowing easy enrichment of existing discrete event simulators with distributed statistical model checking capabilities. MultiVeStA offers: (1) a clean way to integrate discrete event simulators; (2) a language (MultiQuaTEx) to compactly express many systems properties at once; (3) the efficient estimation of the expected values of a MultiQuaTEx expression with respect to n independent simulations, with n large enough to respect a user-specified confidence interval; (4) the plot of the results in a minimal GUI, and the generation of gnuplot input files; (5) a client-server architecture to distribute simulations.

4.4 Novelty and Future work

We introduced new powerful analysis tools based on statistical model checking, to deal consistently with two classes of related problems: system design, a preliminary step towards system control, and the impact of uncertainty on parameters on truth of temporal logic formulae. We set up statistical model checking procedures based on recent machine learning advances, i.e. Gaussian Process-based regression, classification, and optimisation. These methods allow us to approximately solve these two problems in an efficient and statistically consistent way, and they can be extended in many directions.

In the context of the QUANTICOL project, in particular, we will look at devising new statistical methods for the spatio-temporal logics that will be developed for CAS systems. This will include the extension of ideas of quantitative semantics to such logics, and the introduction of novel statistical model checking approaches, possibly based on GP regression and/or classification. We will also look at the design problem in such a spatial context, extending and improving the approach developed during the first year of the project. Another potential application of these ideas is to control problems, in which statistical methods can form the backbone of novel control theoretic approaches that adaptively maintain a system to satisfy its target behaviours. Another line of work will be the application of these methods to the case studies of the project, starting from public transportation models.

5 Towards spatial extensions

This section provides an account of investigation done during the first year of the QUANTICOL project, about logical approaches to the treatment of spatial information. The technical details are described in the report [CLM14]. The material presented in this section is a brief overview of the study of current literature on *spatial logics* presented in [CLM14]. This section also complements Deliverable 2.1, dedicated to spatial models, with a logical perspective. Furthermore, we develop some simple examples related to the smart transport scenario, in order to appreciate the relevance of spatial logics for the QUANTICOL case studies.

5.1 State of the art/base line before start of project

Modal logics, model checking and static analysis enjoy an outstanding mathematical tradition, spanning over logics, abstract mathematics, artificial intelligence, theory of computation, system modelling, and optimisation. However, the *spatial* aspects of systems, that is, dealing with properties of entities that relate to their position, distance, connectivity, reachability in space, have rarely been emphasised in Computer Science. For the QUANTICOL project, it is important to identify logical languages that predicate over spatial aspects, and eventually find methods to certify that a given collective adaptive system satisfies specific requirements in this respect. Below, we provide a brief introduction to spatial logics, their topological (continuous) and closure-based (discrete) interpretation, and the interplay with temporal aspects. We complement the section with an overview of recent developments of spatial logics in Computer Science, most notably *Ambient Logic*, and *Mobile Stochastic Logic*.

[7] Accessible at `code.google.com/p/multivesta/`.

5.1.1 Spatial interpretation of modal logics

Spatial logics have been studied from the point of view of (mostly modal) logics. The field is well developed in terms of descriptive languages and computability/complexity aspects. The development already started with early logicians such as Tarski, who studied possible semantics of classic modal logics, using topological spaces in place of frames. However, the frontier of current research does not yet reach verification problems, and in particular, discrete models are still a relatively unexplored field.

In the first project year, we investigated some relevant current literature dealing with continuous models, and started an analysis of the situation in the case of discrete structures. The interest for the QUANTICOL project in such an analysis comes from the conjecture that properties may be described using the same languages in the continuous, discrete, and relational (classical) case. This should provide an unifying view of temporal and spatial properties which is orthogonal to the kind of models that are taken into account. Our study is intended as a starting point to identify the descriptive languages of interest for the project. The next step will be to cast the studied framework into the realm of discrete and finite structures, and to develop practical verification algorithms.

Spatial logics predicate about entities that are related by a notion of *space*. The *possible worlds* of modal logics become complex mathematical structures. Very often, topological or metric spaces are used. Furthermore, a temporal dimension may be present, and the interplay of space and time gives rise to a rich design space. We have based our study upon the *Handbook of Spatial Logics* [APHvB07], which constitutes an important classification and review effort. Below, we summarise the most important design variables of a reasoning and verification framework based on spatial and temporal logics.

Abstract spatial models To a first approximation, space can be modelled as a discrete or continuous entity (see also Deliverable 2.1, for a detailed classification of spatial models in the literature on collective adaptive systems). The common traits of logical reasoning in the discrete and continuous case ought to be accommodated in a general setting by choosing appropriate abstract mathematical structures. *Topological spaces* are prototypical examples; these are better suited to model the continuous case. On the other hand, discrete structures, such as graphs, are better treated as *closure spaces*, that we discuss in Section 5.1.3. Finer predicates are introduced by *metrics* (or, more generally, *costs*); hereby, models are based on *distance spaces* or *metric spaces*.

Spatial logics Spatial logics predicate on properties of entities located in the space; for example, one may be interested in entities that are *inside*, *outside* or on the *boundary* of regions of space where certain properties hold. Depending on the specific logical language, the entities described can be:

- Points in the space. In this case reasoning has a strongly local flavour. Global properties (e.g., a region of a space not having "holes") can not be expressed.
- Spaces. Global properties can easily be expressed if the point of view is shifted from the behaviour of an individual in a specified setting, to the analysis of several possible global scenarios consisting of all the entities in a given space.
- Regions of space. This approach combines reasoning on multiple entities simultaneously with a focus on the interaction (e.g., overlapping or contact) between areas having different properties.

Metrics and measures Distance-based logics extend topological logics. Formulas are indexed by intervals, which are used as constraints. Metric-topological properties are satisfied by a model if the topological part of the formula is verified, and the constraints are satisfied. For example, one may require that points satisfying a certain property are located at most at a specified distance from each other, or from points characterised by some other property.

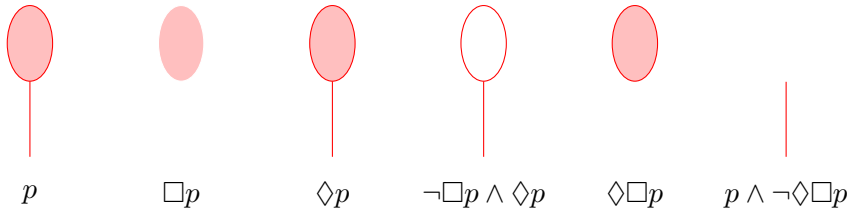


Figure 8: Topological interpretation of formulas over a topo-model.

Spatio-temporal logics The combination of spatial and temporal logics introduces more design variables, especially for what concerns the interplay between the spatial and the temporal component.

5.1.2 Topological spatial logics

We now provide a basic example in order to provide the "look and feel" of spatial logics based on topological models, the so-called *topo-logics*. Namely, we report on the spatial interpretation of modal logic $\mathcal{S}4$ as described in [vBB07]. Several extensions of this basic framework, including distances or geometrical properties, are detailed in the Report [CLM14].

As is usual in logics, we need to describe: the *syntax* of the logic, that is, what are formulas; the *models*, which can *satisfy* or not formulas, and the satisfaction relation. The syntax depends upon a set of proposition letters, that associate properties to points of a space. A formula can be a truth value ($\top$ for *true*, $\bot$ for *false*), a proposition letter p, a formula built using the classical conjunction ($\wedge$), disjunction ($\vee$) and negation $\neg$ operators, or a *modal* formula. Modal formulas are introduced by unary operators $\Diamond$ and $\Box$.

Models of the logic are based on topological spaces. A *topological space* is a pair (X, O) of a set X and a collection $O \subseteq \mathcal{P}(X)$ of subsets of X called *open sets*, such that $\emptyset, X \in O$, and subject to closure under arbitrary unions and finite intersections. A *topological model* $\mathcal{M}$ is a topological space equipped with a *valuation* function $\mathcal{V}$ that associates to each proposition letter p a set of points in X, interpreted as the points where p holds. Satisfaction of a formula is defined at a specific point x. One writes $\mathcal{M}, x \models \phi$ when ϕ holds at point x in model $\mathcal{M}$. In particular, $\Diamond\phi$ holds when, for all open sets o containing x, there is y in o such that $\mathcal{M}, y \models \phi$; $\Box\phi$ holds whenever there is an open set o including x such that for all y belonging to o we have $\mathcal{M}, y \models \phi$. A first example of using the logic is given by the derived operator $\mathcal{B}\phi$, which is a shorthand for $\Diamond\phi \wedge \neg\Box\phi$ and identifies the *boundary* of ϕ, in the sense that the set of points x at which $\mathcal{B}\phi$ holds is the boundary of the set of points where ϕ holds.

We report in Figure 8 the first example from [vBB07]. The topological space here is the two-dimensional Euclidean plane $\mathbb{R}^2$ equipped with the Euclidean topology. The only proposition letter is p and the valuation of p assigns to this property the shape of a "spoon" composed of a line segment and a filled ellipse. Various formulas can denote regions such as the boundary of the spoon, including or excluding the handle, the inner part of the spoon, the whole figure without the handle, etc.

A natural question is what structures are logically equivalent, that is, how fine-grained is the logic. It turns out that logical equivalence coincides with the notion of *topological bisimilarity*, which is an extension of classical bisimilarity between Kripke models. Bisimilarity equates points based on their local properties. Two points are bisimilar when, first of all, the same propositions hold at them in their respective models. Additionally, it is required that, for every open set o_1 on one side, there is a choice of an open set o_2 on the other side, all points of which have a corresponding bisimilar point in o_1. This distinguishes points that are on the boundary of some property from points that are in its interior. Furthermore, the precise shape and size of properties in a model does not affect bisimilarity, which is only driven by the existence of open sets covering each property.

In Figure 9, we depict a model, based on the Euclidean topology of the two-dimensional plane, using colours (red and blue) to depict properties. We use small circles to denote points of interest in the space. In this example, the yellow points are all bisimilar, and so are the green points. The green

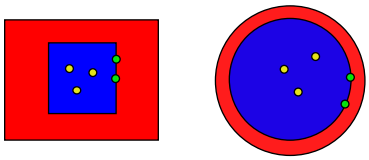


Figure 9: Bisimilar and non-bisimilar points under some predicates (represented by colours in the background). The yellow points are all bisimilar, and so are the green points; however, no green point is bisimilar to any yellow point.

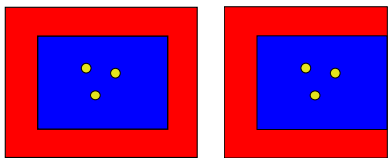


Figure 10: The yellow points (belonging to two different models) are all bisimilar, even if some of them are not completely surrounded by the red property.

points lie on the boundary of the blue property; the yellow points are in its interior. No green point is bisimilar to a yellow point. To see this, choose a yellow point x and a green point y. Note that there exist open sets containing x that are totally contained in the blue property. Let o be such an open set. For each choice of an open neighbourhood of y, on the other hand, there is a point outside the blue property, which does not have any corresponding point in o.

Topo-logics have a strongly local flavour, and are not able to make a distinction between points, driven by properties that are at some distance from them. Consider Figure 10. The yellow points are all bisimilar, even though the red property completely surrounds only some of them. The red property can not affect bisimilarity of yellow points, since there is no physical contact between the two.

Extended logics based on $\mathcal{S}4$ include $\mathcal{S}4_u$, adding operators that reason about the system as a whole, instead of just single points. This is useful to overcome the local flavour of $\mathcal{S}4$, when this is needed. Further logics are obtained by combining $\mathcal{S}4$ and $\mathcal{S}4_u$ with *intervals* in the spirit of quantitative logics; in this case, the models are spaces that are both topological and *metric*, or at least, equipped with a binary, real-valued *distance* function (in metric spaces, such a function is reflexive and obeys the triangular inequality).

Among extended logics, a noteworthy idea is that of using an *until* operator, borrowed from temporal logics, to reason about unbounded regions of space. Formally, a binary operator $\mathcal{U}$ is added to the logic. We have $\mathcal{M}, x \models \phi \,\mathcal{U}\, \psi$ whenever there is an open set o including x such that all the points in o satisfy ϕ and all the points at the boundary of o satisfy ψ. In other words, x lives in an area of the space where ϕ holds, and all points immediately outside such area satisfy ψ. The until operator gives to a logic a different expressive power from that of $\mathcal{S}4$. This is shown in Figure 11, where the formula *blue* $\mathcal{U}$ *red* tells apart the yellow points, which are surrounded by the red property, from the green points, that can reach the "outside" of the red property.

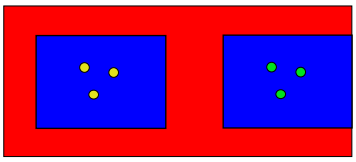
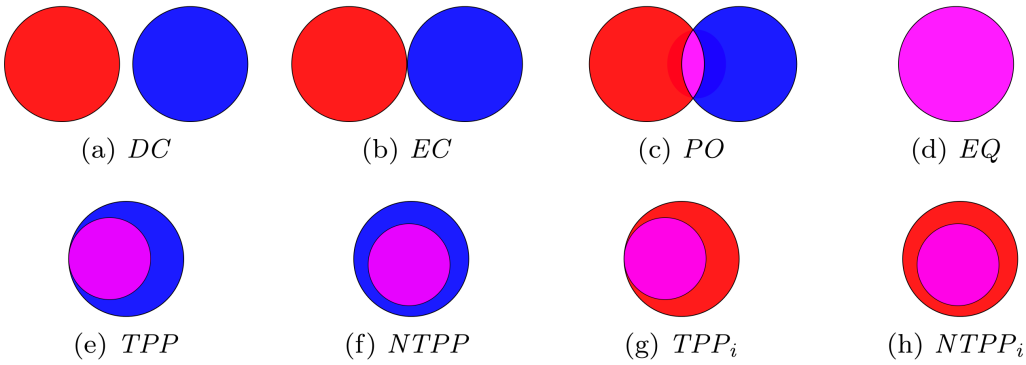


Figure 11: The green points and the yellow points are separated by the formula *blue* $\mathcal{U}$ *red*.



(a) *DC* (b) *EC* (c) *PO* (d) *EQ*

(e) *TPP* (f) *NTPP* (g) TPP_i (h) $NTPP_i$

Figure 12: The eight $\mathcal{RCC}8$ operators; for each operator `bop`, a model is shown where `bop`(*red*, *blue*) is true; violet indicates the overlap of *red* and *blue*.

In the literature on spatial logics, quite some attention has been devoted to *calculi of regions*. Here the object of discourse are so-called *regions*, which are subsets of the points of the space enjoying some regularity property, typically ensuring that the region does not contain "holes" and has a "smooth" boundary. These calculi provide better computational properties, as several cases, which are theoretically difficult to handle, and are not possible in the macroscopic physical world, are ruled out by the definition of a region. Region calculi feature eight binary operators on regions, that depict various situations in which two regions can be. This is illustrated in Figure 12. In particular, the operators denote that the first region is *disconnected* (a), *externally connected* (b), *partially overlapping* (c), *equal* (d), *tangential proper part* (e), *non-tangential proper part* (f), *inverse of tangential proper part* (g), and *inverse of non-tangential proper part* (h), respectively.

5.1.3 Discrete structures and closure spaces

Here we briefly introduce some mathematical foundations that can be used to interpret spatial logics over discrete structures. There is plenty of literature available on treating discrete structures in a similar fashion to topological spaces. In principle, discrete spatial structures could be dealt with as in the continuous case, by defining a topology on top of the points of the structure. However, by doing so, one does not gain much advantage, as the closure operator, responsible for the meaning of the logical operator $\Diamond$, is idempotent in topological spaces. This assumption becomes too stringent for discrete structures. For example, in the case of regular grids, it is natural to interpret closure as the operation of enlarging a set of points by one step (in all possible directions) on the grid. Such interpretation is not idempotent.

In the literature, there are several works that discuss how to study discrete structures, along the lines of the topological treatment of continuous spaces. A well-known subject in this area is *discrete topology*, initiated by Rosenfeld [Ros79] (see [KR89] for a survey). Discrete topology studies properties of *digital images*, that is, multi-dimensional arrays of non-negative numbers, typically interpreted as colors in a regular grid. Despite its name, digital topology is rather graph-theoretical than topological, and most of the topologically inspired notions are defined in an ad-hoc fashion. The same problems have been approached more recently by resorting to a well-known superclass of topological spaces, namely *closure spaces*, where the closure operator is not required to be idempotent (see [Smy95], [Gal99]). A detailed introduction, and a survey of existing results in this respect, is provided by [SW07]. These works are rather exhaustive, but nevertheless very abstract. Often the presented ideas are only applied to graph-like structures that are *symmetric*, where edges can be considered bi-directional. Therefore, we prefer to base the developments of this chapter on the work [Gal03]. This paper further develops the ideas of [Smy95] and [Gal99], enhancing the previous point of view with a theory of closure spaces and continuity principles associated to various discrete structures, including

arbitrary graphs.

By removing the idempotency assumption, *closure spaces* are used in place of topological spaces. A *closure space* is a pair $(X, \mathcal{C})$ where X is a set of points, and $\mathcal{C}$ is an operation on sets, mapping each subset of X to its *closure*. Additional laws hold: the closure of the empty set is empty, the closure of each set A includes A, and the closure of an union of two sets is equal to the union of their closures.

Discrete structures typically come in the form of a graph. A graph whose set of nodes is X is given by binary relation R, describing connectedness of nodes. A closure operator $\mathcal{C}_R$ can be associated with R by letting $\mathcal{C}_R(A)$ be the union of A and all the elements x such that the pair (x, a) is in R. In other words, $\mathcal{C}_R$ augments a set A of nodes by including all the nodes that are source of an arc, whose target is in A. Using this definition, one is able to prove that the obtained closure space $(X, \mathcal{C}_R)$ enjoys *minimal neighbourhoods*, or, equivalently, the closure of a set A is the union of the closure of the singletons composing A. These properties actually single out closure spaces derived from a relation, as they hold in an arbitrary closure space $(X, \mathcal{C})$ if and only if there is some R such that $\mathcal{C} = \mathcal{C}_R$. Spaces enjoying this property are called *quasi-discrete*. Minimal neighbourhoods provide a notion of *step in space* which is missing in arbitrary topological or closure spaces (think for example about Euclidean spaces). Spaces that enjoy this feature are relevant in digital imaging and computer graphics, as they provide the necessary minimal units of representation, such as, pixels of a digital image. In the case of logics and model checking, minimal neighbourhoods permit one to reason locally, visiting relevant parts of the space in discrete steps, starting from a single point. The fact that this form of analysis is correct is also made clear by the characterisation of quasi-discrete spaces as those where the closure of a set is obtained from the union of the closures of the singletons composing it: by this assumption, all the properties that are derived from the definition of closure, locally to a given point, still hold in the whole space.

In a closure space, formula $\Diamond\phi$ is interpreted as the closure of the set of all points satisfying ϕ. In other words, the points satisfying $\Diamond\phi$ are those points that are "one step close" to the points satisfying ϕ. Complex spatio-temporal logics on top of closure spaces are a subject whose investigation has only recently been started; new developments in the field in the context of QUANTICOL (e.g., methods for automated verification) may make a good contribution from the project to the enlarged community of Theoretical Computer Science.

5.1.4 Spatio-temporal logics

In [KKWZ07], the authors extend spatial logics with the ability to predicate at the same time on space, intended as a topological notion, and discrete (linear or branching) time, in the spirit of temporal logics. A number of resulting logics are possible, depending on the choice of spatial operators (such as, those of $\mathcal{S}4$, $\mathcal{S}4_u$, or region calculi), and on the temporal paradigm (typically under the *linear time* or *branching time* assumption). The paper examines some relevant possible combinations.

In classical modal logics, the object of reasoning are *possible worlds* that may have observations on them. Spatio-temporal models enhance this point of view by equipping possible worlds with additional structure, such as a topology or a metric, that may change over time.

The spatial component of such models can be classified as:

- based on *points* of a topological or metric space; a spatial object (say, the interpretation of a formula) coincides with the set of points it occupies.
- based on *regions*; in these models, spatial objects are *regions*, that is, portions of the space, subject to some uniformity or "well-behavedness" conditions.

The temporal aspects of a model can be dealt with as:

- *snapshot-based models*, where the topological space in question evolves "as a whole"; time is described by a total or partial order, and the spatial interpretation of propositions depends upon time;

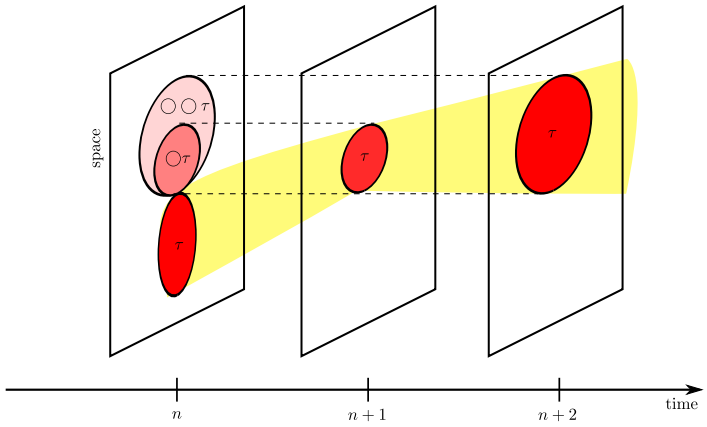


Figure 13: In spatio-temporal logics, a term interpreted at time n may refer to the interpretation of a subterm in some other instant of time.

- *dynamical models*, consisting of a topological space equipped with a total continuous map describing the step-wise and point-wise evolution of a *dynamical system*.

A key difference between most of the traditional temporal logics, and spatio-temporal reasoning is the ability to predicate about the relationship between valuations of certain terms at different times. One may want to express that the position of a certain entity *now* is the same as the position of another entity *tomorrow*. In temporal logics, it is not typical to predicate on such properties. It is rare to encounter temporal logics that can compare the value of variables at different instants of time. In Figure 13, $\bigcirc\tau$ is a spatial term, whose semantics is the interpretation of term τ at the next instant of time. For example, consider operators from region calculi, combined with a temporal operator $\bigcirc$, under the assumption of linear time. Formula $\bigcirc\phi$ is true at a point x and time n, whenever ϕ holds at the same point at time $n+1$. The formula $\bigcirc(EC(p,q)) \leftrightarrow EC(\bigcirc p, \bigcirc q)$ asserts that two regions will be externally connected *in the next step*, if and only if the regions they will occupy in the next step are externally connected *now*.

We avoid going into the technical details of spatio-temporal logics. These require some technical background, and are presented in detail in [CLM14]. Models of the logic are (variants of) so-called *temporal topological models*, which are topological models parameterised over the temporal order of events. For example, in the linear time case, the order of events is the set of natural numbers $\mathbb{N}$, denoting discrete, sequential, linearly ordered *time steps*. For a fixed set of proposition letters, a linear topological model is a topological space, equipped with a family of valuations, one for each natural number, assigning to each proposition letter the set of points where the proposition holds, for each time step.

5.1.5 Spatial Logics in Computer Science and Ambient Logic

The term "spatial logic" has been recently used in Computer Science to refer to logics expressing properties of structured objects, such as processes or data structures. Very often, such spatial logics have been defined as extensions of classical temporal logics. Recent work in this respect stemmed from the areas of: *process calculi*, especially the *π-calculus*, which was designed for expressing properties of mobile processes [Cai04, CC03, LV10], and *mobile ambients* with the related *ambient logic* [CG00]; rewrite theories [BM10, Mes08]; graph-based computational models like bigraphs [CMS07]. In this field, some research has been also carried out on using spatial logics for reasoning about *data structures*, such as *graphs* [CGG02], *heaps* [BDL08, BDL09, Rey02], and *trees* [CGG03, CG04]. The underlying idea is to employ modal (spatial) operators that are the logical counterpart of operators

of the considered description language (such as, a process calculus). For example, in the setting of process calculi with a binary parallel composition operator $\|$, a binary operator $|$ can be defined in the logic as well. Formula $\phi \mid \psi$ holds for process P whenever P is the parallel composition of processes Q and R, such that Q satisfies ϕ and R satisfies ψ. Subtleties (e.g. undecidability issues) arise when one starts considering notions of resources that may be shared possibly under some scopes (such as names in calculi), or that may be created, and further ingredients such as recursion (e.g. fix point operators), or quantifiers (e.g. for any resource, for any fresh/new resource).

The study of spatial operators on processes with resources has been carried on especially in the setting of *Ambient Calculus* and related *Ambient Logic*. Ambient Calculus processes are located and move in *named locations*, that are organised in a hierarchy. Processes may possess *private* names of specific locations, that can be communicated to other processes. Such private locations may be accessed only by processes that know them. The logical counterpart of this construction is the binary modal operator ®, taking as arguments a name x, representing a location in a hierarchical graph, and a formula ϕ, which may contain x. Process P satisfies formula $\phi ® x$ whenever P holds a secret location name l such that P satisfies formula ϕ with x replaced by l. Using this operator, the logic "navigates" trough the levels of the hierarchical structure of locations.

5.1.6 Mobile Stochastic Logic

MoSL [DKL+07] is a logic designed for formally characterizing properties of models specified in StoKLAIM, which is a stochastic extension of KLAIM [DFP98].

A KLAIM *system* is a collection of *sites*, each uniquely identified by its *physical* addresses, with *processes* running and *tuples* stored therein. A *system state* (i.e. a network snapshot) is characterized by a collection of nodes, each located at a specific site. Processes execute *actions*, acting on *local* or *remote* sites. The site at which an action is addressed is specified by means of a *logical* address, which is resolved into a *physical* address by the *allocation environment* of the site where the process executing the action is currently located. KLAIM is based on the tuple-space model of computation: each site is equipped with a tuple *repository* and processes can interact only by means of put-*ing* tuples to repositories or get-*ing*/read-*ing* from repositories tuples matching certain templates. Furthermore, KLAIM processes can be stored/withdrawn/retrieved to/from repositories as well as *spawn* to sites. Finally, they can *create* new sites in the network.

With reference to the classification presented in Deliverable 2.1, the notion of space underpinning KLAIM is **irregular discrete, with aggregation and discrete state** where the connections between locations (sites) are induced by the site allocation environments. More specifically, at a given point of a computation, a site is connected to all and only those sites which are in the range of its allocation environment at that point in the computation. Additionally, the locations and their connections may change during the computation.

In StoKLAIM, each (KLAIM) action is enriched with the specification of the *rate* of a negative exponential distribution which models the *duration* of the execution of the action. Consequently, a CTMC can easily be derived from a bounded StoKLAIM specification[8].

MoSL is a *temporal logic* (i.e. a modal logic addressing the dynamic evolution of systems), which is both *action* and *state* based (i.e. both properties of states and of actions can be addressed by (atomic) propositions); action related propositions have a form of *binding* capability as well. The logic is also a *real-time* and *probabilistic* one (i.e. real time bounds can be expressed on behaviours as well as bounds on the probability that specified events take place) and has explicit treatment of locations where processes run or tuples are stored, but provides no means for reasoning about the connections among these locations; thus the logic offers means for treatment of a limited notion of *space*. Finally, the logic provides modalities for addressing *open* systems.

From the logics point of view, MoSL is based on aCSL [BCH+07], an action-based version of CSL [ASSB00, BHHK03], and MoMo [DL04]. The MoSL atomic proposition $P@\iota$ is satisfied by any

[8] A StoKLAIM specification is bounded iff the LTS defined by the formal semantics is finite.

system state where process P is ready to be executed at site ι[9]. Similarly, $\langle 3\rangle@\iota$ is satisfied by a state if the tuple $\langle 3\rangle$ is stored in the repository of site ι in that state. The classical CSL probabilistic formulas $\mathcal{P}_{\bowtie p}(\varphi)$ and $\mathcal{S}_{\bowtie p}(\Phi)$ are included in MoSL. For instance, $\mathcal{P}_{>0.97}(\varphi)$ states that the probability that *path formula* φ is satisfied in the current state is greater than 0.97, whereas $\mathcal{S}_{<0.3}(\mathcal{P}_{>0.97}(\varphi))$ states that, at equilibrium, less than one third of the system states enjoy the above property. Path formulas are those built using the *next* ($\mathcal{X}$) and *until* ($\mathcal{U}$) CSL operators. For instance, the formula $\Phi\,\mathcal{U}^{\leq 3.14}\Psi$ is satisfied by all those computations where eventually, within 3.14 time units, a state is reached which satisfies Ψ, and all preceding states in the computation (if any) satisfy Φ. Action-based, aCSL-like, extensions of the next and until operators are available as well. For instance, the formula $\Phi\,{}_{\Omega}\mathcal{U}^{\leq 3.14}_{\Delta}\Psi$ enriches the previous one by requiring, in addition, that the state satisfying Ψ must be reached via the execution of an action satisfying Δ, while each preceding state (if any) must be reached from the previous one via an action satisfying Ω. Δ and Ω can specify single actions, like, e.g. a specific put action, as well as sets of allowed actions; variables can also be used in such action specifications and a suitable binding mechanism is provided by the semantics of the logic.

Finally, MoSL offers two more modalities for state-based properties, which are the basic tools for reasoning about *open* systems using the logic. The first one is *production*: a system state satisfies $Q@\iota \leftarrow \Phi$ iff the state satisfies Φ *in case process* Q is executed at site ι. Note that the formula *does not* require Q to be actually present at ι in the state under scrutiny; it can be used to ask questions about *what* happens with a system *if* Q is launched in the current state, and, in particular, whether it is true that Φ holds. Similarly, a *consumption* formula $Q@\iota \rightarrow \Phi$ is satisfied by any state in which Q is ready to be executed at ι and the "remaining" network (i.e. Q's *context*) satisfies Φ. Production and consumption formulas are available also for tuples, with the obvious syntax and meaning.

A model checking algorithm has been designed for StoKLAIM and MoSL [DKL+07] and a model checker has been implemented[10].

5.2 Application to smart transport case studies

In urban transport-related scenarios, one takes into account a geographical map describing the various transport networks, and the fundamental elements of a city, such as roads, squares, rivers, points of attraction. By adding a temporal dimension, entities of several different kinds may move on the map and interact with each other, forming a collective system.

For instance, in the bike sharing case study, one may consider the graph whose nodes are either bike stations or points of interest (e.g., train stations, working places, public offices, entertainment places, and so on). Arcs of the graph connect those nodes that are close to each other. This information defines a quasi-discrete closure space as we explained in Section 5.1.3. Points of the space are the nodes, that may satisfy or not given basic propositions. The valuation of propositions, and the location and number of entities, changes over time, giving rise to a spatio-temporal model of a collective adaptive system. The data defining a spatio-temporal model may be given by the system designer; however, in the setting of the QUANTICOL project, and due to the inherent difficulties in correctly describing collective systems, it is more likely that such data is the outcome of some approximation or simulation method, such as mean-field or fluid-flow approximation. In both cases, the languages of spatial logics permit one to make various kinds of queries (that are then answered by a model checker) spelling out requirements that the model should satisfy. Some possible examples are:

- at every moment, for all empty bike stations, there is a station nearby with some bikes available;
- when there are no bikes available at the train station, it is likely to find some on the path to the city centre;

[9] Note that physical addresses ι are used in formulas, since the latter should not depend on local information.
[10] The model checker is part of the tool set SAM and can be downloaded from `http://rap.dsi.unifi.it/SAM/`

- at every moment, stations with available bikes constitute a boundary for the region of empty stations, so that a user walking in any direction will eventually find a bike;
- when there is a full station, some bikes are removed from it by the bike sharing company within a few time steps;
- the region(s) occupied by stations with more bikes tends to slowly move from the peripherals of the map towards points of interest along time (e.g., because some users actively move bikes from a station to another according to system-specified criteria, in exchange for incentives).

Note that the expressive possibilities of statements about a spatio-temporal model depend upon the considered features in the logic, e.g., probability, named locations, boundaries. As is made clear by the form of the above statements, the goal of a framework comprising logics and verification algorithms is typically to permit flexible specification of system requirements by both system designers and their customers; combined with the approximation methodologies for collective adaptive systems that the QUANTICOL project proposes, this approach ultimately yields the possibility to design and analyse a collective system before deployment.

We conclude this section by emphasizing that, even though most of the spatio-temporal logics we have been investigating during the first year do not cover probabilistic aspects of behaviour or spatial distribution, we think these are important features of the logic in the context of QUANTICOL. Consequently, we plan to address these extensions in the context of this work package.

5.3 Future work

The preliminary investigation on spatial logics performed during the first year is meant to be applied in the context of automated verification (e.g., *model checking*). For tractability reasons, discrete, finite models seem to be preferred to continuous models. Therefore, future work will be initially focused on completing the study of approaches to spatial modelling and spatial logics for discrete structures. In particular, we note that *closure spaces* appear to play a central role in the field. Once established these theoretical foundations, investigation will be directed to automated verification in the presence of discrete spatial information. This will also be extended to handle temporal information, in order to develop a complete framework for automated reasoning about discrete spatio-temporal phenomena. Further extensions that should be studied include quantitative analysis of such systems, using metrics to characterise quantitiative properties of a spatially distributed system (e.g., one may study distances, probabilities, or costs in such a scenario). The role played by space in mean-field/fluid-flow semantics of processes should also be a further subject of investigation.

6 Conclusions and Roadmap

In this deliverable we reported on the progress made during the first year of the project concerning the study of the theoretical foundations of scalable model checking approaches that exploit mean-field and fluid flow approximations. The main aim was, as outlined in the Description of Work of the project for Work Package 3, to develop innovative and efficient fluid model checking techniques that address local, global and mixed stochastic properties of CAS and of individual components that are evolving in the context of a CAS.

Main achievements. Below we briefly summarise the main achievements mentioned in the individual sections:

- Consolidation of the Fluid Model Checking approach by considering two further CSL operators: the steady state and the next-operator [BH13b].

- Lifting local properties to global properties, developing a new method [BL13] based on a second-order fluid approximation known as linear noise approximation, which is a functional version of the central limit theorem.
- Development of a new numeric method in the form of a tool chain for stability analysis of stochastic population models derived from suitable process algebra specifications [BLM13].
- Development of the foundations of a scalable on-the-fly model checking procedure for verifying bounded PCTL formulas based on mean-field and fast simulation results in discrete time. The asymptotic correctness of the procedure has been proven and a prototype implementation of the model checker, FlyFast, has been developed and applied to a number of case studies among which a simplified bike sharing system [LLM13a, LLM13b, LLM13c].
- Development of a new method that allows for the derivation of more accurate error bounds (in probability) on the deviation of the stochastic process from its deterministic limit for a class of deterministic-time Markov population processes [BH13a].
- Development of two new powerful analysis tools based on statistical model checking and recent advances in machine learning, to deal consistently with two classes of related problems: system design (searching relevant parameter spaces) [BBNS13] and the impact of uncertainty on parameters on truth of temporal logic formulae [BS14].
- Literature study on logical approaches to the treatment of spatial information [CLM14].

Relation to other work packages. The work presented in this deliverable relates to other work packages in the following way.

WP1 Emergent Behaviour and Adaptivity. Adaptivity is an important aspect of CAS where the behaviour of the system as a whole may depend on parameter values. Statistical verification techniques, such as those developed in WP3, can form the basis of novel control theoretic approaches that adaptively maintain a system to satisfy its target behaviours. Such control theoretic approaches are studied in WP1.

WP2 Collective Adaptive Behaviour in Space. Among others, WP2 is concerned with finding suitable representations for modelling spatial aspects of CAS. Such representations are an essential ingredient to enrich the models that are currently used in the verification approaches in WP3 and are complementary to the spatial logics that need to be interpreted over the models.

WP4 Language and Design Methodology. Currently the verification approaches in WP3 have been based on elementary specification languages that are suitable to explore and develop the foundations of the verification techniques. However, they need to be enriched and further developed to turn them into suitable specification and modelling languages from a designers point of view. Moreover, such higher-level languages should be suitable to support an appropriate design methodology in which the right verification tools are used at the right moment on suitable models. This requires a close collaboration between WP3 and WP4.

WP5 Model Validation and Tool Support. The theoretical foundations of the developed verification techniques should eventually lead to prototype software tool support. These are essential to be able to validate the techniques on the case studies of the project. Several initial prototypes are being developed and are currently tested, experimented and documented before they can be released to a larger number of users within the project.

Roadmap for the second reporting period. During the first reporting period we have mainly concentrated on the first phase of Task 3.1 (Spatial Stochastic Logics and Scalable Verification). In the second reporting period we plan to start the second phase of Task 3.1 considering spatial model checking and spatial extensions of the scalable model checking approaches. Furthermore, we will consolidate the results of the first phase and work on prototype implementations of the approaches to experiment their application on the case studies of the project.

In the second reporting period also research concerning Task 3.2 (Abstraction Techniques for Scalability Beyond Population Size) and Task 3.3 (Relating Local and Global System Views with Variability Analysis) will be active. In particular, for Task 3.2 we plan to investigate further model-reduction techniques that go beyond that of independence of the population size. Currently, the size of the models depend on the number of local states of objects involved in the system. This number cannot always be assumed to be small, in particular when hierarchical modelling of systems-of-systems is considered.

Within Task 3.3 we plan to study the relationship between the micro and macro views of a CAS, in particular commonalities and variability among the behaviour of components of a CAS will be taken into account. There is the need to develop a rigorous semantics of variability for behavioural models on which further analysis and design techniques can be based. We also plan to study probabilistic and stochastic extensions of such semantics and analysis techniques.

References

[AM11] Musab AlTurki and José Meseguer. PVeStA: A parallel statistical model checking and quantitative analysis tool. In *Algebra and Coalgebra in Computer Science - 4th International Conference, CALCO 2011, Winchester, UK, August 30 - September 2, 2011. Proceedings*, volume 6859 of *LNCS*, pages 386–392. Springer-Verlag, 2011.

[AMS06] Gul A. Agha, José Meseguer, and Koushik Sen. PMaude: Rewrite-based specification language for probabilistic object systems. In *QAPL 2005*, volume 153(2) of *ENTCS*, pages 213–239. Elsevier, 2006.

[APHvB07] Marco Aiello, Ian Pratt-Hartmann, and Johan van Benthem, editors. *Handbook of Spatial Logics*. Springer, 2007.

[ASB+95] Adnan Aziz, Vigyan Singhal, Felice Balarin, Robert K. Brayton, and Alberto L. Sangiovanni-Vincentelli. It usually works: The temporal logic of stochastic systems. In Pierre Wolper, editor, *Computer Aided Verification, 7th International Conference (CAV'07)*, volume 939 of *LNCS*, pages 155–165. Springer, 1995.

[ASSB00] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model checking Continuous Time Markov Chains. *ACM Transactions on Computational Logic*, 1(1):162–170, 2000.

[BBNS13] Ezio Bartocci, Luca Bortolussi, Laura Nenzi, and Guido Sanguinetti. On the robustness of temporal properties for stochastic models. *Electronic Proceedings in Theoretical Computer Science*, 125:3–19, August 2013.

[BCG95] Girish Bhat, Rance Cleaveland, and Orna Grumberg. Efficient on-the-fly model checking for CTL*. In *LICS'95*, pages 388–397. IEEE Computer Society, 1995.

[BCH+07] Christel Baier, Lucia Cloth, Boudewijn R. Haverkort, Matthias Kuntz, and Markus Siegle. Model checking markov chains with actions and state labels. *IEEE Trans. Software Eng.*, 33(4):209–224, 2007.

[BDL08] Remi Brochenin, Stephane Demri, and Etienne Lozes. On the almighty wand. In Michael Kaminski and Simone Martini, editors, *Proceedings of the 17th Annual Conference of the EACSL on Computer Science Logic (CSL'08)*, volume 5213 of *LNCS*, pages 323–338. Springer, 2008.

[BDL09] Remi Brochenin, Stephane Demri, and Etienne Lozes. Reasoning about sequences of memory states. *Annals of Pure and Applied Logic*, 161(3):305–323, 2009.

[BEE+10] R. Bakhshi, J. Endrullis, S. Endrullis, W. Fokkink, and B. Haverkort. Automating the mean-field method for large dynamic gossip networks. In *QEST 2010*, pages 241–250. IEEE Computer Society, 2010.

[BGH08] Jeremy T. Bradley, Stephen T. Gilmore, and Jane Hillston. Analysing distributed internet worm attacks using continuous state-space approximation of process algebra models. *J. Comput. Syst. Sci.*, 74(6):1013–1032, 2008.

[BH12] Luca Bortolussi and Jane Hillston. Fluid model checking. In M. Koutny and I. Ulidowski, editors, *CONCUR 2012*, volume 7454 of *LNCS*, pages 333–347. Springer-Verlag, 2012.

[BH13a] Luca Bortolussi and Richard A. Hayden. Bounds on the deviation of discrete-time Markov chains from their mean-field model. *Perform. Eval.*, 70(10):736–749, 2013.

[BH13b] Luca Bortolussi and Jane Hillston. Checking individual agent behaviours in Markov population models by fluid approximation. In Marco Bernardo, Erik P. de Vink, Alessandra Di Pierro, and Herbert Wiklicky, editors, *SFM*, volume 7938 of *Lecture Notes in Computer Science*, pages 113–149. Springer, 2013.

[BHHK03] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-Checking Algorithms for Continuous-Time Markov Chains. *IEEE Transactions on Software Engineering. IEEE CS*, 29(6):524–541, 2003.

[BHLM13] Luca Bortolussi, Jane Hillston, Diego Latella, and Mieke Massink. Continuous approximation of collective system behaviour: A tutorial. *Performance Evaluation*, 70(5):317 – 349, 2013.

[Bil95] Patrick Billingsley. *Probability and measure*. Wiley, New York, USA, 3rd edition, 1995.

[BL08] M. Benaïm and J. Y. Le Boudec. A class of mean field interaction models for computer and communication systems. *Performance Evaluation*, 65(11-12):823–838, 2008.

[BL13] Luca Bortolussi and Roberta Lanciani. Model checking Markov population models by central limit approximation. In *Quantitative Evaluation of Systems 2013*, volume 8054 of *Lecture Notes in Computer Science*, pages 123–138. Springer-Verlag, 2013.

[BLM13] L. Bortolussi, D. Latella, and M. Massink. Stochastic Process Algebra and Stability Analysis of Collective Systems. In R. De Nicola and C. Julien, editors, *Coordination Models and Languages*, volume 7890 of *LNCS*, pages 1–15. Springer-Verlag, 2013. DOI: 10.1007/978-3-642-38493-6_1, ISSN: 0302-9743, ISBN: 978-3-642-38492-9 (print), 978-3-642-38493-6 (on line).

[BM10] Kyungmin Bae and José Meseguer. The linear temporal logic of rewriting maude model checker. In Peter Csaba Ölveczky, editor, *Proceedings of the 8th International Workshop on Rewriting Logic and its Applications (WRLA'10)*, volume 6381 of *LNCS*, pages 208–225. Springer, 2010.

[BS14] Luca Bortolussi and Guido Sanguinetti. Smoothed model checking, 2014. Online at `http://arxiv.org/abs/1402.1450`; accessed 7-March-2014.

[BD13] Lubŏs Brim, Milan Cěska, Sven Drăzan, and David Safŕanek. Exploring parameter space of stochastic biochemical systems using quantitative model checking. In *Computer Aided Verification 2013*, volume 8044 of *LNCS*, pages 107–123. Springer, 2013.

[Cai04] L. Caires. Behavioral and spatial observations in a logic for the π-calculus. In I. Walukiewicz, editor, *Proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures (FOSSACS'04)*, volume 2987 of *LNCS*, pages 72–87. Springer, 2004.

[CC03] L. Caires and L. Cardelli. A spatial logic for concurrency (part I). *Information and Computation*, 186(2):194–235, 2003.

[CES86] Edmund M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, 8(2):244–263, 1986.

[CG00] Luca Cardelli and Andrew D. Gordon. Anytime, anywhere: Modal logics for mobile ambients. In *Proceedings of the 30th SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'00)*, pages 365–377, 2000.

[CG04] L. Cardelli and G. Ghelli. TQL: A query language for semistructured data based on the ambient logic. *Mathematical Structures in Computer Science*, 14(3):285–327, 2004.

[CGG02] L. Cardelli, Ph. Gardner, and G. Ghelli. A spatial logic for querying graphs. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP'02)*, volume 2380 of *LNCS*, pages 597–610. Springer, 2002.

[CGG03] L. Cardelli, Ph. Gardner, and G. Ghelli. Manipulating trees with hidden labels. In A. Gordon, editor, *Proceedings of the 6th International Conference on Foundations of Software Science and Computation Structures (FOSSACS03)*, volume 2620 of *LNCS*, pages 216–232. Springer, 2003.

[CLM14] V. Ciancia, D. Latella, and M. Massink. Logics of Space and Time. Technical Report TR-QC-01-2014, QUANTICOL, 2014.

[CLR09] Augustin Chaintreau, Jean-Yves Le Boudec, and Nikodin Ristanovic. The age of gossip: spatial mean field regime. In John R. Douceur, Albert G. Greenberg, Thomas Bonald, and Jason Nieh, editors, *SIGMETRICS/Performance*, pages 109–120. ACM, 2009.

[CMS07] Giovanni Conforti, Damiano Macedonio, and Vladimiro Sassone. Static bilog: a unifying language for spatial structures. *Fundamenta Informaticae*, 80(1-3):91–110, 2007.

[CVWY92] C. Courcoubetis, M. Vardi, P. Wolper, and M. Yannakakis. Memory-efficient algorithms for the verification of temporal properties. *Form. Methods Syst. Des.*, 1(2-3):275–288, 1992.

[DFM13] Alexandre Donzé, Thomas Ferrère, and Oded Maler. Efficient robust monitoring for STL. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification 2013*, volume 8044 of *LNCS*. Springer, 2013.

[DFP98] Rocco De Nicola, Gian Luigi Ferrari, and Rosario Pugliese. Klaim: A kernel language for agents interaction and mobility. *IEEE Trans. Software Eng.*, 24(5):315–330, 1998.

[DG08] Robin Donaldson and David Gilbert. A model checking approach to the parameter estimation of biochemical pathways. In Monika Heiner and Adelinde M. Uhrmacher, editors, *Computational Methods in Systems Biology*, volume 5307 of *LNCS*, pages 269–287, 2008.

[DKL+07] Rocco De Nicola, Joost-Pieter Katoen, Diego Latella, Michele Loreti, and Mieke Massink. Model checking mobile stochastic logic. *Theor. Comput. Sci.*, 382(1):42–70, 2007.

[DL04] R. De Nicola and M. Loreti. A modal logic for mobile agents. *ACM Transactions on Computational Logic. ACM Press*, 5(1):79–128, 2004.

[DM10] Alexandre Donźe and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In *Formal modeling and analysis of timed systems 2010*, volume 6246 of *LNCS*, pages 92–106. Springer, 2010.

[DN08] R.W.R. Darling and J.R. Norris. Differential equation approximations for Markov chains. *Probability Surveys*, 5:37–79, 2008.

[FG13] C. Fricker and N. Gast. Incentives and redistribution in bike-sharing systems, 2013. Online at `http://arxiv.org/abs/1201.1178`; accessed 17-September-2013.

[FP09] Georgios E. Fainekos and George J. Pappas. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science*, 410:4262–4291, 2009.

[Gal99] Antony Galton. The mereotopology of discrete space. In *Spatial Information Theory: Cognitive and Computational Foundations of Geographic Information Science, International Conference COSIT '99, Stade, Germany, August 25-29, 1999, Proceedings*, volume 1661 of *Lecture Notes in Computer Science*, pages 251–266. Springer, 1999.

[Gal03] Antony Galton. A generalized topological view of motion in discrete space. *Theor. Comput. Sci.*, 305(1-3):111–134, 2003.

[GG10] Nicolas Gast and Bruno Gaujal. A mean field model of work stealing in large-scale systems. In Vishal Misra, Paul Barford, and Mark S. Squillante, editors, *SIGMETRICS 2010*, pages 13–24. ACM, 2010.

[GHLP06] G. Guirado, T. Hérault, R. Lassaigne, and S. Peyronnet. Distribution, approximation and probabilistic model checking. In *PDMC 2005. LNCS, vol. 135*, pages 19–30. Springer, 2006.

[GM11] Stefania Gnesi and Franco Mazzanti. An abstract, on the fly framework for the verification of service-oriented systems. In Martin Wirsing and Matthias M. Hölzl, editors, *Results of the SENSORIA Project*, volume 6582 of *LNCS*, pages 390–407. Springer, 2011.

[HB10] Richard A. Hayden and Jeremy T. Bradley. A fluid analysis framework for a Markovian process algebra. *Theor. Comput. Sci.*, 411(22-24):2260–2297, 2010.

[HBC13] Richard A. Hayden, Jeremy T. Bradley, and Allan Clark. Performance specification and evaluation with unified stochastic probes and fluid analysis. *IEEE Trans. Software Eng.*, 39(1):97–118, 2013.

[Hen98] Thomas A. Henzinger. It's about time: Real-time logics reviewed. In Davide Sangiorgi and Robert de Simone, editors, *CONCUR'98 Concurrency Theory*, number 1466 in Lecture Notes in Computer Science, pages 439–454. Springer Berlin Heidelberg, January 1998.

[HHZ11] Ernst Moritz Hahn, Holger Hermanns, and Lijun Zhang. Probabilistic reachability for parametric Markov models. *International Journal on Software Tools for Technology Transfer*, 13(1):3–19, 2011.

[Hil05] Jane Hillston. Fluid flow approximation of PEPA models. In *Quantitative Evaluation of Systems, 2005*, pages 33–42. IEEE, 2005.

[HJ94] Hans Hansson and Bengt Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.

[HLMP04] T. Hérault, R. Lassaigne, F. Magniette, and S. Peyronnet. Approximate probabilistic model checking. In *VMCAI04. LNCS, vol. 2937*, pages 73–84. Springer, 2004.

[Hol04] Gerard J. Holzmann. *The SPIN Model Checker - primer and reference manual*. Addison-Wesley, 2004.

[HSB12] Richard A. Hayden, Anton Stefanek, and Jeremy T. Bradley. Fluid computation of passage-time distributions in large Markov models. *Theoretical Computer Science*, 413(1):106–141, 2012.

[JCL+09] S. K. Jha, E. M. Clarke, C. J. Langmead, A. Legay, A. Platzer, and P. Zuliani. A bayesian approach to model checking biological systems. In *Proceedings of the 7th International Conference on Computational Methods in Systems Biology, CMSB 2009*, volume 5688 of *Lecture Notes in Computer Science*, pages 218–234, 2009.

[JLS12] Cyrille Jegourel, Axel Legay, and Sean Sedwards. Cross-entropy optimisation of importance sampling parameters for statistical model checking. In *Computer Aided Verification 2012*, volume 7358 of *LNCS*, pages 327–342. Springer, 2012.

[KKLW07] J.-P. Katoen, D. Klink, M. Leucker, and V. Wolf. Three-valued abstraction for continuous-time markov chains. In *Proceedings of the 19th International Conference on Computer Aided Verification, CAV 2007*, volume 4590 of *Lecture Notes in Computer Science*, pages 311–324. Springer, 2007.

[KKWZ07] Roman Kontchakov, Agi Kurucz, Frank Wolter, and Michael Zakharyaschev. Spatial logic + temporal logic = ? In Aiello et al. [APHvB07], pages 497–564.

[KNP04] M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic Symbolic Model Checking using PRISM: A Hybrid Approach. *STTT*, 6(2):128–142, 2004.

[KR89] T. Yung Kong and Azriel Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48(3):357–393, 1989.

[KRd12] A. Kolesnichenko, A. Remke, and P.-T. de Boer. A logic for model-checking of mean-field models. Technical Report TR-CTIT-12-11, http://doc.utwente.nl/80267/, 2012.

[KRd13] A. Kolesnichenko, A. Remke, and P.-T. de Boer. A logic for model-checking of mean-field models. In *DSN'13*. IEEE, 2013.

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

[LLM13a] D. Latella, M. Loreti, and M. Massink. On-the-fly Fast Mean-Field Model-Checking. In M. Abadi and A. Lluch Lafuente, editors, *Trustworthy Global Computing, 8th International Symposium, TGC 2013, Buenos Aires, Argentina, August 30-31, 2013, Revised Selected Papers*, volume 8358 of *LNCS*. Springer, 2013. (To appear on 31 March, 2014).

[LLM13b] D. Latella, M. Loreti, and M. Massink. On-the-fly fast mean-field model-checking: Extended version. *CoRR*, abs/1312.3416, 2013.

[LLM13c] D. Latella, M. Loreti, and M. Massink. On-the-fly PCTL Fast Mean-Field Model-Checking for Self-organising Coordination - Preliminary Version. Technical Report TR-QC-01-2013, QUANTICOL, 2013.

[LMM07] Jean-Yves Le Boudec, David McDonald, and Jochen Mundinger. A generic mean field convergence result for systems of interacting objects. In *QEST07*, pages 3–18. IEEE Computer Society Press, 2007. ISBN 978-0-7695-2883-0.

[Lot24] A. J. Lotka. *Elements of Mathematical Biology*. Williams and Wilkins Company, 1924.

[LV10] Étienne Lozes and Jules Villard. A spatial equational logic for the applied π-calculus. *Distributed Computing*, 23(1):61–83, 2010.

[Mes08] José Meseguer. The temporal logic of rewriting: A gentle introduction. In Pierpaolo Degano, Rocco De Nicola, and José Meseguer, editors, *Concurrency, Graphs and Models, Essays Dedicated to Ugo Montanari on the Occasion of His 65th Birthday*, volume 5065 of *LNCS*, pages 354–382. Springer, 2008.

[MFS+11] M. A. Montes de Oca, E. Ferrante, A. Scheidler, C. Pinciroli, M. Birattari, and M. Dorigo. Majority-rule opinion dynamics with differential latency: A mechanism for self-organized collective decision-making. *Swarm Intelligence*, 5(3–4):305–327, 2011.

[MN04] Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In Yassine Lakhnech and Sergio Yovine, editors, *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems 2004*, volume 3253 of *LNCS*, pages 152–166. Springer, 2004.

[RBFS11] Aurélien Rizk, Grégory Batt, François Fages, and Sylvain Soliman. Continuous valuations of temporal logic specifications with applications to parameter optimization and robustness measures. *Theoretical Computer Science*, 412(26):2827–2839, 2011.

[RdBSH14] D. Reijsbergen, P.T. de Boer, W. Scheinhardt, and B. R. Haverkort. On hypothesis testing for statistical model checking. *Submitted to The International Journal on Software Tools for Technology Transfer (STTT)*, 2014.

[Rey02] J. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings of the 17th IEEE Symposium on Logic in Computer Science (LICS'02)*, pages 55–74. IEEE Computer Society, 2002.

[Ros79] Azriel Rosenfeld. Digital topology. *The American Mathematical Monthly*, 86(8):621–630, 1979.

[RW06] Carl Edward Rasmussen and Christopher K. I Williams. *Gaussian processes for machine learning*. MIT Press, Cambridge, Mass., 2006.

[SHB10] A. Stefanek, R. A. Hayden, and J. T. Bradley. A new tool for the performance analysis of massively parallel computer systems. In *QAPL 2010. EPTCS, vol. 28*, pages 159–181, 2010.

[SKKS12] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, May 2012.

[Smy95] M.B. Smyth. Semi-metrics, closure spaces and digital topology. *Theoretical Computer Science*, 151(1):257 – 276, 1995. Selected Papers of the Workshop on Topology and Completion in Semantics.

[SV13] Stefano Sebastio and Andrea Vandin. MultiVeStA: Statistical Model Checking for Discrete Event Simulators. To appear in the proceedings of ValueTools 2013, ACM Digital Library, 2013.

[SVA05] Koushik Sen, Mahesh Viswanathan, and Gul A. Agha. Vesta: A statistical model-checker and analyzer for probabilistic systems. In *QEST*, pages 251–252. IEEE, 2005.

[SW07] Michael B. Smyth and Julian Webster. Discrete spatial models. In Aiello et al. [APHvB07], pages 713–798.

[TDGH12] Mirco Tribastone, Jie Ding, Stephen Gilmore, and Jane Hillston. Fluid rewards for a stochastic process algebra. *Software Engineering, IEEE Transactions on*, 38(4):861–874, 2012.

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

[vBB07] Johan van Benthem and Guram Bezhanishvili. Modal logics of space. In Aiello et al. [APHvB07], pages 217–298.

[vK07] N. G. van Kampen. *Stochastic processes in physics and chemistry*. Elsevier, Amsterdam; Boston; London, 2007.

[Vol26] V. Volterra. Fluctuations in the abundance of a species considered mathematically. *Nature*, 118:558 – 560, 1926.

[Wal45] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, June 1945.

[YKNP04] H. Younes, M. Kwiatkowska, G. Norman, and D. Parker. Numerical vs. statistical probabilistic model checking: An empirical study. In K. Jensen and A. Podelski, editors, *Proc. 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'04)*, volume 2988 of *LNCS*, pages 46–60. Springer, 2004.

[YS02] Håkan L. S. Younes and Reid G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *Computer Aided Verification 2002*, volume 2404 of *LNCS*, pages 223–235, 2002.

[YS06] Håkan L. S. Younes and Reid G. Simmons. Statistical probabilistic model checking with a focus on time-bounded properties. *Information and Computation*, 204(9):1368–1409, September 2006.

[ZPC10] Paolo Zuliani, Andŕe Platzer, and Edmund M. Clarke. Bayesian statistical model checking with application to Simulink/Stateflow verification. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pages 243–252. ACM, 2010.