On Performance Analysis of Collective Adaptive Systems ⋆

Maurizio Murgia[1], Riccardo Pinciroli[1], Catia Trubiani[1], and Emilio Tuosto[1]

Gran Sasso Science Institute, Italy
{maurizio.murgia, riccardo.pinciroli, catia.trubiani, emilio.tuosto}@gssi.it

Abstract We compare a traditional approach based on Petri Nets and suitable for to the analysis of non-functional properties of collective adaptive systems (CAS) with a recent one advocating the use of behavioural abstractions. We use a case study based on a scenario involving autonomous robots to discuss the relative merit of the approaches.

1 Introduction

Increasingly *Collective adaptive systems* (CAS) crop up in many application domains, spanning critical systems, systems assisting humans during their working or daily live activities, smart cities. A paradigmatic example is the use artificial autonomous agents in rescue contexts that may put operators lives at stake.

eM: citare qualcosa?

These agents execute in a cyber-physical context and are supposed to exhibit an *adaptive* behaviour. This adaptation should be driven by the changes occurring in their operational environments, besides by the changes in the local state of computation of components, "collectively taken". Also, the *global* behaviour of CAS should *emerge* from the *local* behaviour of its components. Let us explain this considering the coordination of a number of robots patrolling some premises to make sure that aid is promptly given to human operators in case of accidents.

A plausible local behaviour of each robot can be: (1) to identify accidents, (2) to assess the level of gravity of the situation (so to choose an appropriate course of action), (3) to alert the rescue centre and nearby robots (so to e.g, divert traffic to let rescue vehicles reach the location of the accident more fastly), and (4) to ascertain how to respond to alerts from other robots (e.g., if already involved in one accident or on a low battery, a robot my simply forward the alter to other nearby robots). Note that robots' behaviour depend on the physical environment (tasks **??????**) as well as its current state in **??**.

A possible expected global behaviour is that robots try maximise the patrolled area while trying to avoid remaining isolated and to minimise the battery consumption. It is worth remaking that the global behaviour is not typically *explicitly* programmed; it rather should be the resultant of the combination of the

⋆ Research partly supported by the EU H2020 RISE programme under the Marie Skłodowska-Curie grant agreement No 778233. Work partly funded by MIUR project PRIN 2017FTXR7S *IT MATTERS* (Methods and Tools for Trustworthy Smart Systems) and by MIUR project PRIN 2017TWRCNB *SEDUCE* (Designing Spatially Distributed Cyber-Physical Systems under Uncertainty).

behaviour of the single components. For instance, when designing the algorithm for the roaming one could assume that a robot does not move towards an area where there already a certain number of robots.

This paper applies behavioural specifications to the quantitative analysis of CAS. Using a simple, yet representative scenario, we show how to use behavioural specifications and use them to study non-functional properties of CAS (emergent) behaviour. More precisely, we use Petri nets and behavioural types to model a robot scenario recently proposed in [?]. These two models are rather different and their application allows us to make a comparison between the two approaches.

eM: todo

Outline.

2 A Robot Scenario

eM: tbc

We describe the scenario that we will use for our analysis and discussion.

HIC SUNT LEONES CAS = CPS?

If CAS $\neq$ CPS
then [?] not designed for CAS AND [?] not designed for changes
else ???
fi

We consider the analysis presented in [?] to assess the performance characteristics of architectural patterns in a cyber-physical system in presence of dynamic physical changes. The analysis conducted in this paper shows that by considering a set of models it is possible to measure the impact of architectural patterns and dynamic space changes on the performance of cyber-physical systems. This type of analysis suggests that in this domain it is useful to factor performance in at design time. The approach proposed in [?] is new and it hinges on generalised stochastic petri nets (GSPN) [?].

eM: weak

In this paper we apply such approach to CAS and study. Also, we study the relative merits of the approach in [?] and the one proposed in [?] advocating the combination of behavioural specifications and queuing networks [?] for performance analysis of CAS. More precisely we address the following two research questions:

RQ1 To what extent the approaches in [?] and in [?] support performance-aware design of CAS?

RQ2 How the features of the approaches in [?] and in [?] compare?

2.1 A Petri Net Model

2.2 An Behavioural Sepcification Model

The *interactions* for proposed in [?] are, in their most general form, of the form

$$\mathsf{A}_{|}\rho \xrightarrow{e \quad e'} \mathsf{B}_{|}\rho' \tag{1}$$

and they are meant to capture the following intuition:

> "*any* agent, say A, *satisfying* ρ generates an expression e for *any* agents satisfying ρ', dubbed B, provided that expression e' *matches* e."

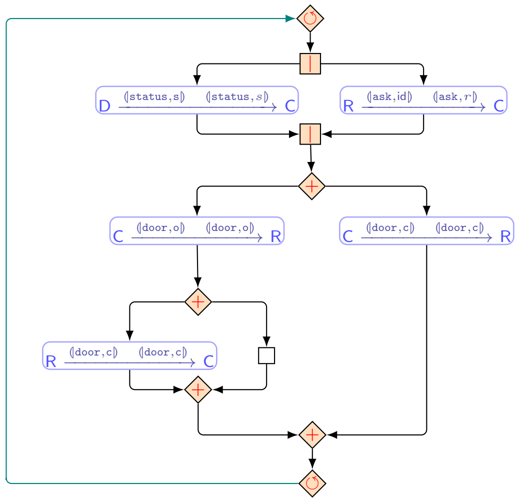
The conditions ρ and ρ' predicate over some application-dependent *attributes* exposed by components. These conditions are used to address communication partners. More precisely, any component whose attributes satisfy ρ (resp. ρ') can act as senders (resp. receiver). The payload of an output is a tuple of values e to be matched by the sender with e'; when e and e' match, the effect of the communication is that the variables in e' are instantiated with the corresponding values e.

As said, a send operation targets components satisfying a given *predicate* on such attributes. For instance, a suitable attribute for the robot scenario considered in **??** is the position of the robot so that a robot at position p can send an alert to all robots at a certain distance from p. Likewise, each potential receiver performing a receive operation eventually gets messages depending on its receiving predicate ρ'. Any other message is disregarded.

Role names A and B in (**??**) are pleonastic: they are used just for succinctness and may be omitted for instance writing $\rho \xrightarrow{e \quad e'} \mathsf{B}_{|}\rho'$ or $\rho \xrightarrow{e \quad e'} \rho'$. Also, we abbreviate $\mathsf{A}_{|}\rho$ with A when ρ is a tautology.

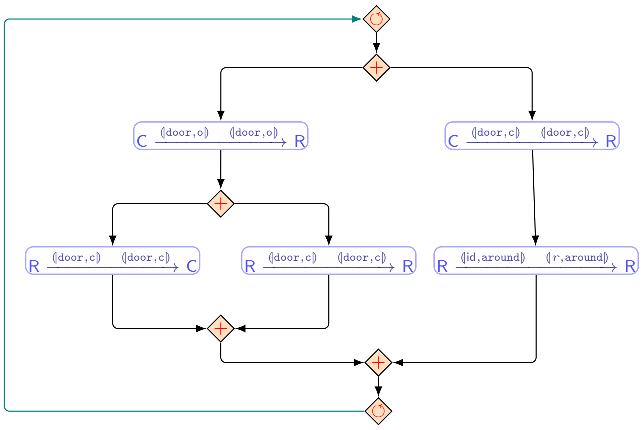
Interactions are the basic elements of an algebra of protocols [?] featuring iteration as well as non-deterministic and parallel composition. In the diagrams (**??**) and (**??**) below (which we discuss later), the ◇-gates mark the entry and exit points of loops. Likewise, the ◇-gates mark the branching and merging points of a non-deterministic choice. Finally, we use □-gates to mark forking and joining points of parallel protocols.

Centralised



(2)

Distributed



(3)

3 Quantitative Analysis

4 Conclusions, Related & Future Work