MAXIM-GPRT: A Simulator of local Schedulers, Negotiations, and Communication for Multi-Agent Systems in General-Purpose and Real-Time Scenarios

Giuseppe Albanese[3], Davide Calvaresi[1,2], Paolo Sernani[3], Fabien Dubosson[2], Aldo Franco Dragoni[3], and Michael Schumacher[2]

[1] Scuola Superiore Sant'Anna,Pisa, Italy
[2] University of Applied Sciences Western Switzerland, Sierre, Switzerland
[3] Università Politecnica delle Marche, Ancona, Italy
peppe-a@live.it, d.calvaresi@sssup.it,
{p.sernani,a.f.dragoni}@univpm.it, michael.schumacher@hevs.ch

Abstract. In safety-critical scenarios, the compliance with strict-timing constraints is mandatory. This demo presents a simulator named MAXIM-GPRT enabling the analysis of the behaviors produced by Multi-Agent Systems (MAS) composed of both General-Purpose (GP) and Real-time (RT) algorithms. Therefore, MAXIM-GPRT is crucial to prove that current MAS cannot provide timing guarantees, nor guarantee correct behaviors in the worst case scenario. However, adopting and adapting models and algorithms from RT systems, such a compliance, can be achieved.

Keywords: MAS, Timing-Reliability, Deadline Missing-Rate, MAS Simulator

1 Introduction

In a society increasingly interconnected, the new generation of systems are evolving towards the *Internet of Everything* (IoE). In the IoE era, a multitude of distributed electronic devices, continuously interacting with the environment and collecting data from each other, couple virtual and real domains by reconciling so-called Cyber-Physical System (CPS) solutions. However, CPS are increasingly employed in safety-critical scenarios. Their behavior is therefore required to be correct in terms of both result and deliberation time (i.e., dependable and reliable behaviors). Unfortunately, current frameworks supporting the development of Multi-Agent Systems (MAS) only adopt General-Purpose (GP) algorithms pursuing the *best-effort approaches* [3], which give no means to offer any timing guarantee, not off-line nor on-line. Hence, there is a need for understanding and evaluating the system behavior in respecting timing constraints and giving timing guarantees according to given inputs and systems setups.

2 Main Purpose

According to Calvaresi et al. [4], current multi-agent platform and applications are incapable of enforcing the compliance with strict timing constraints (impossibility of providing any guarantee about the system behaviors in the worst-case scenario). Therefore, the adoption of MAS is hampered, excluding significant application scenarios such as "safety-critical environments".

The main reasons for this lack of real-time (RT) satisfiability in MAS originate from current theories, standards, and technological implementations. In particular, traditional internal agent schedulers, communication middlewares, and negotiation protocols have been identified as co-factors inhibiting real-time compliance [2].

The main purpose of this demonstration is to present MAXIM-GPRT, a simulator developed to study MAS performances, possible risks, and failures in both GP and RT scenarios. MAXIM-GPRT allows to integrate MAS with the RT theory that, by studying computing systems that must guarantee bounded and predictable response times [1], provides the means to evaluate timing constraints and guarantees. In particular, the schedulers and negotiation protocols used in the most known agent frameworks [3], and a selection of schedulers typical of real-time systems, have been implemented inside the simulator and made modular. Table 1 describes the possible outputs provided by the simulator. Table 2 details the configurable elements characterizing MAXIM-GPRT.

Table 1: Simulation results

Id	Indicator	Description
I1	Deadline Miss Ratio (DMR)	number of deadlines missed by a task in a given simulated time.
I2	Lateness (LT)	extra time required by a task missing its deadline to complete.
I3	Response Time (RTM)	amount of time required to complete a given task.

Table 2: Configurable parameters

Id	Parameter	Description
P1	Number of agents	number of agents participating in the simulation.
P2	Agent utilization	load of the agent's CPU (see Section 3.1).
P3	Agent knowledge	set of tasks an agent is able to execute.
P4	Agent task-set	set of running tasks.
P5	Agent Services	set of tasks an agent might execute on demand.
P6	Agent Needs	set of tasks an agent needs, but it is unable to execute.
P7	Tasks models	typology of running tasks.
P8	Tasks utilization	load of a single task (see Section 3.1).
P9	Negotiation prot.	mechanisms used to negotiate task execution.
P10	Heuristics	policies used by agents to select possible contractors and to award them.

3 Demonstration

MAXIM-GPRT relies on the OMNET++ framework[4], and is composed of an arbitrary amount (P1) of simple modules organized in a fully connected network with configurable channels (e.g., type of connection, communication delays, and connected modules). Such modules embody the agents in a community. They have the same structure with customizable attributes and capabilities (see Figure 1 (a)). An additional agent is by default part of the community. It is the Directory Facilitator (DF). Such a concept is adopted from the FIPA standard for agent management[5] and is in charge of providing a list of agents willing to offer given services. To enable completely autonomous executions and dynamics, the parameters listed in Table 2 have to be defined before starting the simulation. Such parameters (mainly organized in XML files) can be defined by hand or automatically, exploiting a tool to generate them according to given ranges and distributions.

[4] https://omnetpp.org/

[5] http://www.fipa.org/specs/fipa00023/

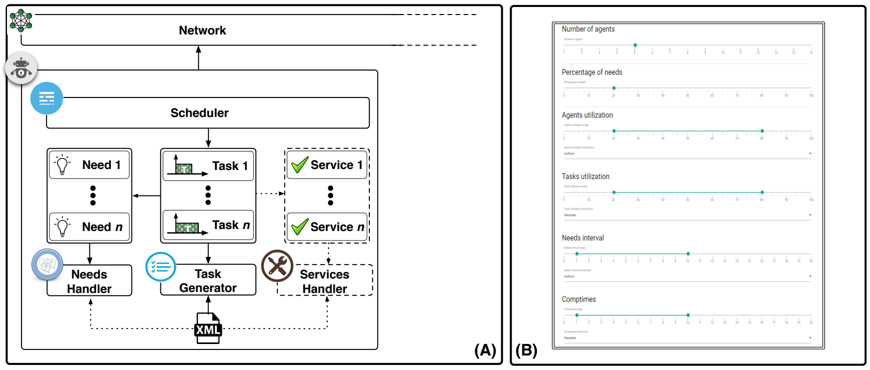


Fig. 1: (a) Agent composition, (b) parameters generator web interface.

3.1 MAXIM-GPRT: Possible Setups

Figure 1 (b) shows the two interfaces (command line and web) to setup the parameters in Table 2 with the following values:

[P1]: an integer value $(x \geq 1)$; [P2]: a real value $(0 < x \geq 1)$; [P3]: a set of tasks characterized by: id, executor, demander, computation time[6], residual computation time, arrival time[6], relative deadline, period[6], number of executions, first activation time, last activation time, public flag, server id (XML format); [P4]: a sub-set of the tasks (marked as activable) in P3. [P5]: a sub-set of tasks (marked as public) in P3. [P6]: a sub-set of tasks that an agent is unable to execute, and that might be part of someone else P5. Each need is characterized by a starting time and a number indicating its required execution(s); [P7]: periodic, periodic in an interval, aperiodic; [P8]: a real value[6] $(0 < x \geq 1)$; [P9]: Contract Net Protocol and Reservation-Based Negotiation Protocol; [P10]: (H1) select first agent in the list, (H2) select a random agent, (H3) select a random subset of agents, (H4) select the best offer according to a cost function;

P1 and P9 are the only parameters valid for the entire community. The remaining parameters can be set for every agent singularly.

3.2 MAXIM-GPRT: Demostration of simulation analysis

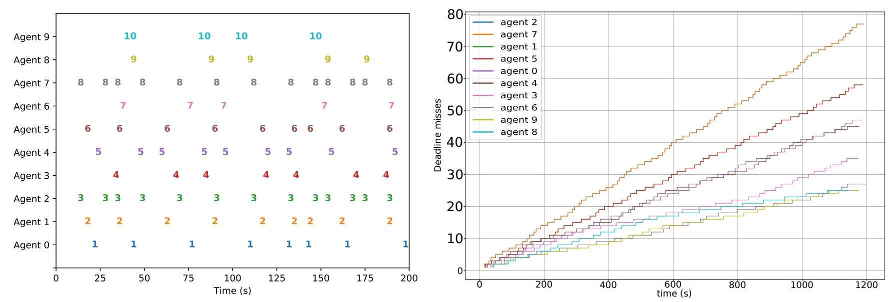
The proposed demo shows how simulation analysis of deadline miss and task-set scheduling can be performed on a MAS. Every simulation can be run "event per event", at a given (arbitrarily fast) speed, or completed in "one click". At the completion of every simulation, MAXIM-GPRT provides reports composed of:

DMR: for every occurrence, it contains time and details of the tasks that missed their deadlines, the agent's utilization factor, and the running tasks at that given point in time. *LT:* it integrates the DMR info reporting the extra time needed to complete the execution by tasks missing their deadlines. *RTM:* it provides the response time of all the executed tasks (including those missed their deadlines). *Statistics:* it contains the total number of deadlines checked, and the number of deadlines missed for each agent in the platform during a given simulation.

Concerning the indicators presented in Table 1, we developed a script to plot the following information collected in the simulation log. The demo will show the available

[6] Values computed according to a uniform probability distribution.

graphs that are: *(i)* the plot of given task(s) missing its deadline(s) in a given agent(s) in the total or a given time interval (see Figure 2a), *(ii)* the plot of the cumulative deadline missed by every agent that took part in a given simulation over the total or a given time interval (see Figure 2a), *(iii)* plots aggregating data collected by the logs and the reports obtain after several simulations in similar and different setups[7].



(a) DMR for 10 agents, [0-200s].

(b) DMR for 10 agents, [0-1200s].

Fig. 2: Results of the deadline miss analysis.

4 Conclusions

Pursuing the timing-reliability in MAS, the proposed work provides a tool, MAXIM-GPRT, to analyze deadline miss ratio, response-time, and lateness of the agent's behaviors (see Table 1) employing GP and RT algorithms (see Table 2). In light of the findings produced by MAXIM-GPRT, it can be concluded that to be able to employ MAS in scenarios demanding the compliance with strict-timing constraints, the adoption and adaption of real-time scheduling models is crucial. The development of the MAXIM-GPRT simulator revealed to be strategic to prove that a task missing its deadline can be the output of several factors such as the agent utilization factor, single task utilization factor, and task-set composition. As an ongoing work, we are studying a qualitative evaluation in terms of response time and lateness (in case of deadline miss) between FCFS, RR, and EDF (CBS when employing sporadic tasks).

References

1. Buttazzo, G.: Hard real-time computing systems: predictable scheduling algorithms and applications, vol. 24. Springer Science & Business Media (2011)
2. Calvaresi, D., Appoggetti, K., Lustrissimini, L., Marinoni, M., Sernani, P., Dragoni, A.F., Schumacher, M.: Multi-agent systems negotiation protocols for cyber-physical systems: Results from a systematic literature review. In: Proceedings of 10th International conference on agents and artificial intelligence (2018)
3. Calvaresi, D., Marinoni, M., Lustrissimini, L., Appoggetti, K., Sernani, P., Dragoni, A.F., Schumacher, M., Buttazzo, G.: Local scheduling in multi-agent systems: getting ready for safety-critical scenarios. In: Proceedings of 15th European Conference on Multi-Agent Systems. Springer (Dec 2017)
4. Calvaresi, D., Marinoni, M., Sturm, A., Schumacher, M., Buttazzo, G.: The challenge of real-time multi-agent systems for enabling iot and cps. In: Proceedings of the International Conference on Web Intelligence. pp. 356–364 (2017), http://doi.acm.org/10.1145/3106426.3106518

[7] Due to space limitations only *(i)* and *(ii)* have been shown.