Published in Proceedings of the 18th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2020) : Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness, L'Aquila, Italy, 7-9 October 2020, which should be cited to refer to this work.
DOI: 10.1007/978-3-030-49778-1_30

SEAMLESS: Simulation and Analysis for Multi-Agent System in Time-Constrained Environments

Davide Calvaresi[1], Giuseppe Albanese[1], Jean-Paul Calbimonte[1], and Michael Schumacher[1]

University of Applied Sciences and Arts Western Switzerland HES-SO, Sierre, Switzerland
{name.surname}@hevs.ch

Abstract. The correctness of a system operating in time-constrained scenarios leverages on both precision and delivery time of its outcome. This paper presents SEAMLESS, a system enabling the design, simulation, and in-depth analysis of Multi-Agent Systems (MAS). In particular, SEAMLESS allows defining in detail the agents' knowledge (set of tasks it might execute), needs (set of tasks to be negotiated), local scheduler (execution of the task-set), negotiation protocols, possible communication delays, and heuristics related to the parameters mentioned above. This tool is pivotal in the strive to study and realize real-time MAS.

1 Introduction

In time-critical scenarios, compliance with deadlines and response times is imperative. Focusing on the ontological differences between the *execution* of "code" and "process" *acting* in the real world, dealing with *time* (in both virtual and real environments) is a fundamental requirement crucially entangled with *deadlines*, *precedence*, *priority*, and *constrained resources*. In the last decades, MAS researchers developed several agent-oriented models, languages and frameworks, although none of them is able to reason and operate *in* time — thus being incapable of considering and dealing explicitly with strict timing constraints [3]. Hence, MAS have been confined in narrowed application domains, rarely employed in the real-time compliant systems. To foster MAS in widening their application domains, there is the need for deeply understanding the behavior of their characterizing algorithms, protocols, and inputs with respect to time. SEAMLESS has been designed as a design, simulation, and evaluation system to fill this gap.

2 Objectives and Requirements

In MAS, the incapability of complying with strict-timing constraints stems from current theories, standards, and technological implementations of the MAS foundations. In particular, traditional agent internal schedulers, communication middleware, and negotiation protocols are co-factors inhibiting real-time compliance [3]. Unlike RTS-algorithms, general purpose (GP) algorithms do not allow mathematical means to provide off-/on-line timing guarantees. Thus, to understand the effects of employing a given algorithm within a specific scenario, we set the following goals: (i) to enable the creation of customizable input (e.g., task-sets and needs); (ii) to allow the customization of the system parameters (see Table 2); (iii) to provide a detailed representation (both graphical and logs) of the simulation's outcome for analytical purposes. Furthermore, we set specific requirements to facilitate the use of SEAMLESS: (i) Accessibility: The system is available online, through a multi-dev responsive Web-interface, not requiring local installation or deployment. (ii) Portability: the software is docker-contained; (iii) OpenSource: The code will be opened in accordance with the funding agency policies.

3 Demonstration

SEAMLESS provides several indicators to characterize the simulation of the designed MAS (Table 1). The parameters can refer to the overall MAS (G), to each agent (S), or both, aggregating values (A) or the trends spanning over the entire simulated time (P).

Table 1: Simulation indicators

Id	Indicator	Description	A/P	G/S
I1	Deadline Miss Ratio (DMR)	number of deadlines missed by a task.	A	S
I2	Lateness (LT)	extra time required by a task missing its deadline to complete	P	S
I3	Response Time (RTM)	amount of time required to complete a given task	P	S
I4	Utilization Factor (UF)	Utilization of the Agent's processor	P	G,S
I5	Potential UF (PUF)	Agent's UF variance subject to pending negotiations	P	S
I6	Acceptance Ration (AR)	Ratio among negotiated and accepted task	A	G,S
I7	Task-set Exectution (TE)	Graphical representation of the task schedule	P	S

3.1 SEAMLESS Architecture

The components of the SEAMLESS architecture (Figure 1) are organized in four Docker containers: client (React), server, backend-simulator (Omnetcpp and Node.js Express), database (Postgres), and Redis) orchestrated by Docker compose. The interface enables to register, login, manage the user's details, manage the simulation (create, delete, and edit existing ones), and provides analysis support (i.e., graphical analysis tools and formatted simulation logs). The backend-simulator is an improved and extended version of MAXIM-GPRT [1], which executes simulations of MAS characterized by many new parameters, protocols, algorithms, and heuristics (see Table 2). Task-sets, needs, and parameters can be configured via the web-interface by the user. When the setup is completed, they are organized in JSON files and fed to MAXIM-GPRT. When the execution is completed, the outcome is stored into the DB and linked to the user profile.

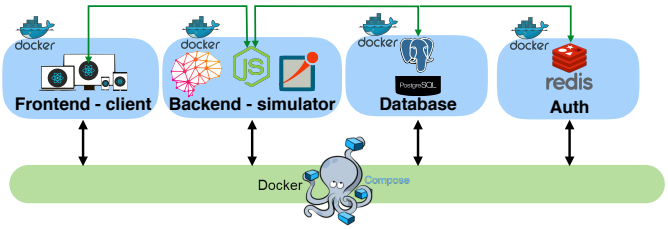


Fig. 1: SEAMLESS Architecture.

3.2 SEAMLESS Setup

Table 2 describes the main parameters that can be set via web-interface characterizing the inputs and the system setup. Figure 2 shows their graphical representation. These parameters, as detailed in Table 2, allow indicating the number of agents (P1), task execution capabilities (P2, including id, computation time, arrival time, deadline, period, activation time, etc.), task set (P3), on-demand services (P4), agent needs (P5, e.g. tasks needed to be executed, including their release time), etc. Most of the parameters are fully customizable (including P7-P9), while others have preset options, e.g., the local scheduler (P11) and the negotiation protocol (P10, e.g., Contract Net, Reservation-Based Negotiation, Reservation-Based Negotiation Plus, English Auction, Dutch Auction).

Table 2: Configurable Parameters

Id	Parameter	Description
P1	Number of agents	number of agents participating in the simulation.
P2	Agent knowledge	set of tasks an agent is able to execute.
P3	Agent task-set	set of running tasks.
P4	Agent Services	set of tasks an agent might execute on demand.
P5	Agent Needs	set of tasks an agent needs, but it is unable to execute.
P6	Tasks models	typology of running tasks.
P7	Agent utilization	load of the agent's CPU [2]
P8	Tasks characterization	the features can be computational time, period, deadline, the number of executions, demander, executor, release time, server, isPublic, isActive.
P9	Network delay	customizable min and max value to define a communication delay among the agents
P10	Negotiation prot.	mechanisms used to negotiate (bid and award) task(s) execution.
P11	Local scheduler	algorithm scheduling the agent tasks/behaviors and related mechanisms (e.g., server) [2].
P12	Heuristics	policies used by agents to select possible contractors and to award them.

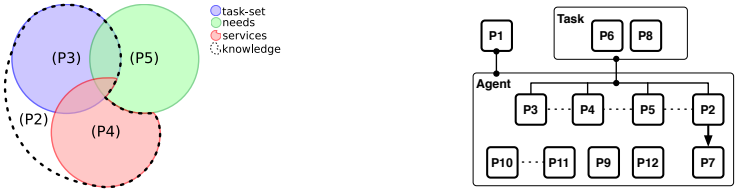


Fig. 2: (a) Venn diagram of P2-5. (b) Graphical representation of a Scenario.

3.3 SEAMLESS Simulation Analysis

SEAMLESS proposes the possibilities of downloading verbose formatted logs and graphical representations of (I1 to I7) and general statistics. Due to space limitations, we can only show a few of them. Figure 3 shows the evolution of I4 of each agent representing a snapshot of the distribution of the computations in the system. Figure 4 shows the actual behavior of P11. the most granular representation of the exectution of two periodic tasks (T0 and T5) and two aperiodic (read and write messages) served by the respective servers (S200 and S100). Figure 5a shows I3 of three tasks (P11 - FIFO) and (P11 - CNET). In particular, it is possible to see how the interference provoked by the release of task t_4 impacts on t_0. Moreover, releasing t_8, the task-set becomes unstable, causing the divergence of its response time. Under real-time assumption (EDF+CBS and RBN), the potential P7 is computed by the schedulability test to verify the impact on P7 of possibly accepting a task execution. Figure 5b shows P7 (purple line) and the potential P7 utilization (black line). The agent receives a two proposals in a few seconds. Making its computation the agent decides to bid positively to both. In turn, the bid are accepted, and P7 gets aligned with the potential P7.

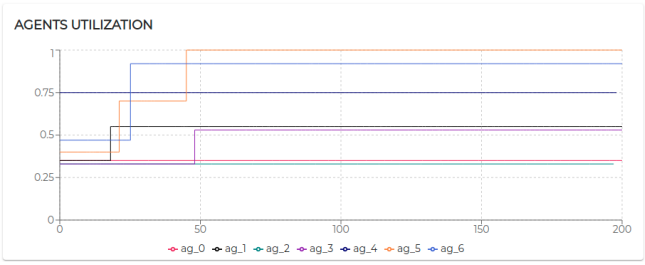


Fig. 3: Utilization (y-axes) of 7 agents over 200 simulated seconds (x-axes).

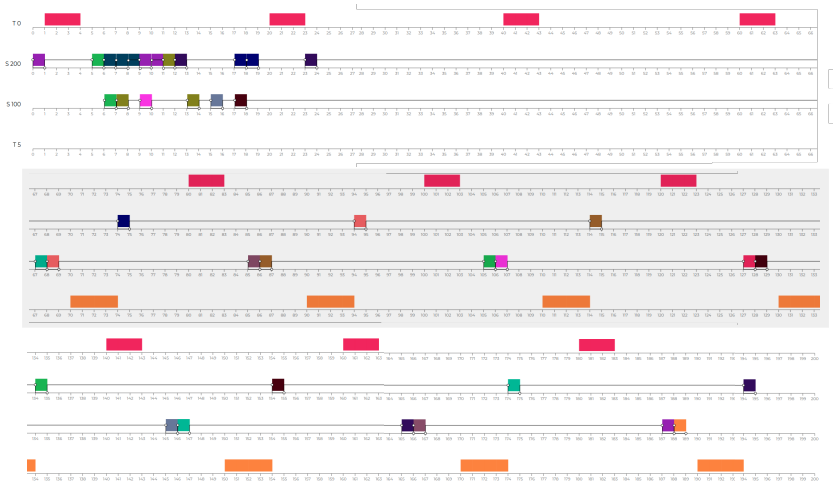
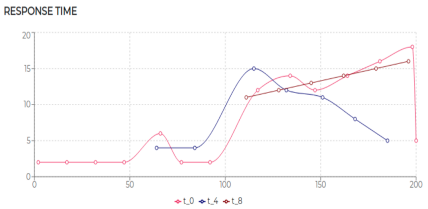
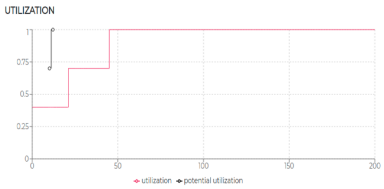


Fig. 4: Graphical representation of the simulated scheduling activity of the Earliest Deadline First (EDF) combined with the Constant Bandwidth Server (CBS) over 200s.



(a) Response Time of tasks t_0, t_4, t_8.



(b) Potential and effective Agent's utilizaion.

4 Conclusions

SEAMLESS copes with the need for understanding and evaluating the system behavior w.r.t. deadlines of both GP and RT-algorithms. Hence, the proposed system allows to analyze several time-critical performance indicators (see Table 1) employing GP and RT algorithms (see Table 2). SEAMLESS is strategic in supporting the design and analysis of time-critical MAS, enabling future adoption of RT-compliant systems.

References

1. Albanese, G., Calvaresi, D., Sernani, P., Dubosson, F., Dragoni, A.F., Schumacher, M.: Maxim-gprt: A simulator of local schedulers, negotiations, and communication for multi-agent systems in general-purpose and real-time scenarios. In: International Conference on Practical Applications of Agents and Multi-Agent Systems. pp. 291–295. Springer (2018)
2. Buttazzo, G.: Hard real-time computing systems: predictable scheduling algorithms and applications, vol. 24. Springer Science & Business Media (2011)
3. 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. ACM (2017)