IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 5, 2019, accepted March 6, 2019, date of publication March 13, 2019, date of current version April 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904760

# Dynamic Reconfiguration of Intelligence for High Behaviour Adaptability of Autonomous Distributed Discrete-Event Systems

**HAMZA GHARSELLAOUI**[1,2,3,4] **AND MOHAMED KHALGUI**[1,4]

[1]School of Intelligent Systems Science and Engineering, Jinan University (Zhuhai Campus), Zhuhai 519070, China
[2]Khurmah University College, Taif University, Taif 26571, Saudi Arabia
[3]National Engineering School of Carthage (ENICarthage), Carthage University, 2035 Ariana, Tunisia
[4]National Institute of Applied Sciences and Technology (INSAT), Tunis Cedex 1080, Tunisia

Corresponding authors: Hamza Gharsellaoui (gharsellaoui.hamza@gmail.com) and Mohamed Khalgui (khalgui.mohamed@gmail.com)

This work was supported by the School of Intelligent Systems Science and Engineering, Jinan University (Zhuhai Campus), Zhuhai, China.

**ABSTRACT** This paper deals with the intelligence adaptation of distributed real-time embedded control systems when scenarios of reconfiguration happen in their hardware or software level. The reconfiguration process is a composition of controllers reconfiguration by adding/deleting or updating tasks and intelligence reconfiguration by adding/deleting and updating the rule base. The system architecture is composed of an application layer implemented as the real-time periodic tasks and an intelligence layer for autonomous and adaptive control behavior. In this research work, a rule-based system is used as the artificial intelligence component, where we propose to optimize the inference process by splitting the rule base into two sub-bases; the effective one and the general meta base. With this facility, the coordination in decision making for the distributed platform and system Quality of Service (QoS) in complex and robust system implementation should be faced by new strategies and correct policies. In this sense, we propose a new protocol for the coordination in the two levels of the reconfiguration process to get correctness in the system results. Dealing with performance, we propose to supervise the intelligence QoS of the whole distributed system. Also, we present the correctness of the coordination between the different decisions by the field of the coordination factor. An implementation of the paper contribution with Drools as an integrated rule-based system framework to RTDroid and a discussion of the inferring time and the memory consumption are presented in this paper.

**INDEX TERMS** Real-time embedded control systems, intelligence reconfiguration, rule-based system, intelligence QoS, coordination factor, intelligence impact.

## NOMENCLATURE AND GLOSSARY

| | |
|---|---|
| QoS | Quality of Service. |
| DIS | Distributed Intelligent System. |
| AI | Artificial Intelligence. |
| RBS | Rule-Based System. |
| RTOS | Real-Time Operating System. |
| WCET | Worst Case Execution Time. |
| ERB | Effective Rule Base. |
| GMB | Generic Meta-Base. |
| MAS | Multi-Agent System. |
| IF | Intelligence Factor. |
| IR | Is Running. |
| CF | Coordination Factor. |
| RHS | Right Hand Side. |
| LHS | Left Hand Side. |
| CM | Coordination Matrix. |
| CoRDIS | Coordination in Reconfigurable Distributed Intelligent System. |
| CSs | Control Systems |
| DCs | Distributed Control Systems |

The associate editor coordinating the review of this manuscript and approving it for publication was Nasim Ullah.

## I. INTRODUCTION

Control systems (CSs) are designed to perform functions to regulate physical processes such as automotive, avionics and industrial automation. Distributed control systems (DCSs) which consist of a set of CSs, are constantly evolving in term of flexibility and agility [1], [6]–[15] and their complexity is increasing according to their fields of application [2], [3]. There are two categories of dynamic reconfiguration: (i) Manual reconfiguration executed by users, and (ii) Automatic reconfiguration executed by an intelligent agent that can be a physical resource (robot, machine, ...) [4], [5] or a logical resource (scheduler).

With the grow of intelligent systems exploitation in both manufacturing and digital economy, different challenges are presented for the optimization of the quality of services. Various hardware design and flexible software architecture enhance the autonomous and adaptive control behaviour in distributed embedded control platforms. Such systems suffer of limited resources constraints (low power consumption, low processor frequency, low memory...) and give importance to time exploitation and to the process of coordination between the different devices in the feasibility of such systems. In this context, authors of our community focus their works on the system architecture design by proposing novels refinement and injecting different layers to optimize resources allocation, system portability and to deploy adaptive architectures. The reconfiguration technology has belong a greatest solution for adaptive system to face environment changes and to perform autonomous behaviour. On one hand, the hardware reconfiguration technology aims to change the hardware design of the system using programmable hardware [30]. On the other hand, the dynamic software reconfiguration technology adapts systems automatically to the environment changes by modifying their parameters, deleting some tasks or messages and switching priorities [9], [27]. In [9] work, a software/hardware/network task is considered as a real-time task which is essentially characterized by its: (i) Period $T$ which representing the interval between two consecutive instances, (ii) Computation time $C$ being the worst case execution time (WCET), (iii) Deadline $D$ being the time by that the task must be accomplished. All those solutions help tasks or messages to meet their real-time constraints in critical situations and help also systems to continue performing results with a low-power consumption [9]. Those two solutions satisfy human invisibility constraint in pervasive computing and they reinforce techniques for context awareness computing by getting specific control behaviour for the DIS depending on system execution context [20]. Wang *et al.* [9] and Khalgui *et al.* [31] focus their research work in the reconfiguration of periodic and aperiodic tasks which are triggered by simple events. Nevertheless, no one in our community [2]–[5], [14], [46]–[48] has address the problem of the reconfiguration of intelligent tasks which are triggered by a rule engine as an AI component. The AI implementation in embedded control systems is focused on the definition of self-tuning controllers, supervisory expert control and generation and management of application software for control systems [23]. Nalepa and Ziecik [24], developed an Embedded Prolog Platform for embedded hardware. The work depicted in [25] proposes an implementation of rule-based system in a microcontroller for a smart home. Davis *et al.* [21] present a taxonomy and a survey in the application of AI techniques for adaptive distributed real-time embedded control system. Authors show in this work that a variety of AI techniques may be applied to adaptive DIS. Unlike this survey didn't propose an implementation of adaptive AI techniques in DIS with a low memory constraint and response time discuss. Dealing with adaptive and autonomous system in AI authors developed machine learning methods [29] to contain the environment changes. This technique generates and adds new rules for the rule base to resolve requests which were previously not supported by the system [28]. The learning methods invoke algorithms when, the coming requests/questions are denied from the system, or to obtain new knowledge and rules mining from existing data set or from experiences/examples [28], [29]. The current paper deals with an original issue which is the implementation of the artificial intelligence that describes the flexibility of the system by a set of configurations presented in the RHS of each rule. In other words, the automatic concurrent reconfigurations of intelligence in distributed discrete-event real-time systems under inference response time and memory constraints. This reconfiguration process is a composition of controllers reconfiguration by adding/deleting or updating tasks and intelligence reconfiguration by adding/deleting and updating the rule base. Indeed, an inference rule is a trigger to activate/deactivate a set of periodic tasks or/and to reconfigure task's parameters. The LHS of rules are set of functional conditions depending on a set of defined sensor's measurements values in the working memory of the RBS. This formal description presents the reconfiguration of controllers with a rule-based system taken in this research work as the AI component of the system architecture and using the match-resolve-act cycle [22]. Complex reconfigurable systems having greatest number of configurations should be optimized by minimizing the number of inference rules in function of system execution context. Works in [36] and [37] propose techniques for rule base partitioning nevertheless they don't propose a methodology for rule base reconfiguration. In this position, our previous works [32], [33] propose a new architecture for intelligence and controllers reconfiguration in AI-based adaptive control system with intelligence QoS, real-time and low-memory feasibility. Those contributions have made satisfiable results for centralized system adaptation when changes occur in the environment and give us the idea to explore the adaptation features of the rule base in a distributed platform to optimize the inference response time and the hardware resources management. In current system architecture, services and process controls are distributed between a set of connected embedded devices. These systems [12] encode several types of tasks (software, hardware and network) to handle several types of reconfiguration scenarios under classically functional and extra-functional constraints (i.e. security and real-time [16]). These scenarios are assumed to be off-line or run-time concurrent operations to improve the performance or save the system on the occurrence of software/hardware faults and also the networking threats [14], [19]. Dealing with synchronization and coordination, many research works have used different protocols and techniques to ensure those constraints. MAS is used for this field in Khalgui's work [31] by providing a reconfiguration agent for each device of the DIS to manage the reconfiguration process and a coordination agent to ensure the feasibility of distributed reconfiguration scenarios. This solution didn't study the specificity of the rule based systems where interconnected devices may lead to critical problems and inconsistent decisions such as incoherent rule base. Therefore, in this work, we deal with the following problems: i) Coordination making for distributed platform and system Quality of Service (QoS) in complex and robust system implementation, ii) Management of the intelligence QoS of the whole distributed system, and iii) Management of the correctness of the coordination between the different decisions by the field of the coordination factor under inferring time and memory consumption constraints. The originality of our work is presented as an architectural software design to optimize the system performance in memory consumption, response time and taking account the QoS for the whole AI-based adaptive DIS. Indeed, we propose to devise the rule base of the system into two parts; one defined as the ERB and the other is the GMB. This later contains all the defined rules for all the possible system controller's configurations. The main contribution of our work is the proposition of a new protocol to coordinate between the different devices in a distributed platform. An intelligence QoS factor depicted as $IF_G$ and a coordination factor CF are presented in this research work. We defend the paper's contribution by a set of simulations to watch the IF, the time inferring and the memory consumption of the proposed architecture compared to a static system where it has only a direct access to GMB without the proposed restriction of ERB and the reconfiguration process. It makes the following contributions: 1) A new useful protocol for coordination making for distributed platform and system Quality of Service (QoS) in complex and robust system implementation, 2) New solutions for guaranteeing the management of the intelligence QoS of the whole distributed discrete-event system, 3) New solutions for constructing dynamically the correctness of the coordination between the different decisions by the field of the coordination factor under inferring time and memory consumption constraints, and 4) Innovative proposition of a new protocol to coordinate between the different devices in a distributed platform. We note that these contributions are relevant to any dynamic reconfiguration of intelligence for high behaviour adaptability of autonomous distributed discrete-event systems that have the considered architecture under the considered constraints of this paper. To the authors knowledge, until this research no one in our community deals with the dynamic software reconfiguration of the rule base in DIS. Our methodology which is composed of different steps is original since it gives system Quality of Service (QoS) in complex and robust system implementation, response time and memory guarantees that no one in all related works deals with. The remainder of the paper is organized as follows. Section 2 describes the intelligent system architecture and formalization. Section 3 presents the contribution and the new protocol for the coordination in DIS. In section 4, we discuss the proposed solution by an implementation of the proposed architecture with a simulation scenarios to supervise the variation of IF, CF, the inferring time and the memory allocation size after each reconfiguration scenario. Section 5 concludes the paper.

## II. FORMALIZATION

In this section, we present the system architecture and its formalization to explore the problem of our research work.

### A. DISTRIBUTED INTELLIGENT SYSTEM

In the proposed approach of this paper, the distributed intelligent system is composed of k connected devices $\sum = \{\Theta_1, \Theta_2, \ldots, \Theta_k\}$. The software architecture of each device $\Theta_i$ is composed of three layers; *Controllers*, *Intelligence* and *RTOS* where the controllers component are periodic/sporadic real-time tasks and the intelligence component is a rule-based system as shown in Fig. 1. This later is implemented as a sporadic task to ensure both real-time feasibility with an RTOS scheduling policy of the whole software architecture and to perform an adaptive behaviour by reconfiguring the real-time tasks at run-time. Indeed, two types of tasks are managed by each device in DIS, RTask for Reconfigurable Tasks and STasks for Static Tasks and which are non-reconfigurable one (as some of OS Tasks, or user-defined non-reconfigurable tasks). The hypothesis of this research work is that all STasks are independent from RTask.

We denote $\tau_i(t) = \{T_i^1, \ldots, T_i^n\}$ the set of RTasks and $\Gamma_i(t) = \{\gamma_i^1, \ldots, \gamma_i^m\}$ the set of rules of each device $\Theta_i$.

The temporal descriptions of RTasks are defined as: $T_i^j$ $(R_i^j, C_i^j, P_i^j, P_{max}^{i,j}, D_i^j)$ where $R_i^j$ is the arrival time, $C_i^j$ is the WCET, $P_i^j$ is the period, $P_{max}^{i,j}$ is the maximum user-defined value for reconfiguring the period of a task and $D_i^j$ is the deadline. A sporadic server is implemented for the RBS execution having as a capacity $Cap_i$, a period $Per_i$ and a deadline $Ded_i$. Indeed, the scheduler of the system is implemented using RM [34] and the system is feasible when: $\sum_{j=1}^{n} \frac{C_i^j}{P_i^j} + \frac{Cap_i}{Per_i} \leq (\sum_{i=1}^{n} n*(2^{\frac{1}{n}} - 1)) - \sum_{j=1}^{x} \frac{C_i^j}{P_i^j}$ where x is the cardinality of all STasks; this last equation is simplified by the following statement: $\sum_{j=1}^{N} \frac{C_i^j}{P_i^j} \leq (\sum_{i=1}^{n} n*(2^{\frac{1}{n}} - 1)) - X$ where N = n + 1. In our previous work [33], we proposed a novel approach to reconfigure tasks parameters by updating the period of tasks to their $P_{max}$ when real-time feasibility halts or critical battery charge status denoted as $E_{cr}$ and based on a new parameter the intelligence impact.

Let $S_i = (S_i^1, S_i^2, \ldots, S_i^v)$ denotes the sensors connected to each device $\Theta_i$. The values of sensors measurements $S_i^{k\in v}$.val are the entries of the RBS working memory. Using the match-resolve- act cycle, the RBS $F_i$ decides which rule to fire and the whole AI component of each device $\Theta_i$ of $\sum$ can be

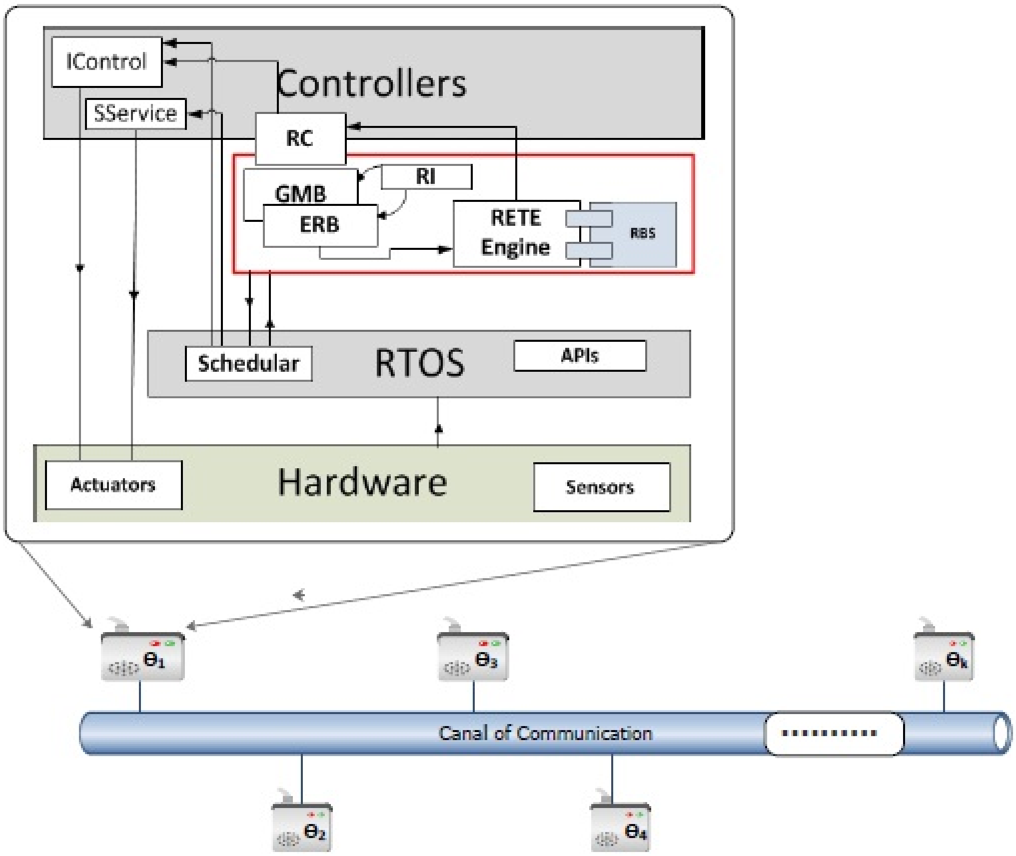


**FIGURE 1.** DIS architecture.

formalized as follows:

$$F_i = (\tau_i^{Bef}, \gamma_i^j) \rightarrow \tau_i^{Aft} \quad (1)$$

$$\gamma_i^j = EC_i^j \rightarrow \{\{T_i^j, j \in [1, n]\}, \{T_i^j.P = T_i^j.P_{max}, j \in [1, n]\}\} \quad (2)$$

$$EC_i^j = S_i^k.val \text{ operel Ctel} \quad (3)$$

$$\text{U} > (\sum_{i=1}^{n} n * (2^{\frac{1}{n}} - 1)) - \text{X}|\text{E} \leq E_{cr}|EC_i^j; \quad \text{k} = 1..\text{v} \quad (4)$$

$$\text{operel} = < | > | \preceq | \geq | = |! = \quad (5)$$

$$\text{U} = \sum_{j=1}^{N} \frac{C_i^j}{P_i^j} \quad (6)$$

### B. COHERENT RECONFIGURATION OF DIS

In this research work, we propose to implement the features of the dynamic software reconfiguration as an architectural optimization to $\Sigma$. Indeed, once a moment $t$ there is specific set of tasks running on the high-level of $\Theta_i$ defined by the set $\tau_i(t)$ and its cardinality $n_i(t)$ also specific set of rules $\Gamma_i(t)$ having as cardinality $m_i(t)$. First, we define $\tau_i(t)$ as a finite state machine with the following formalization:

$$SMAI_i = (\chi_i, \vartheta_i, \psi_i) \quad (7)$$

where the alphabet $\chi_i$ is the set of possible running tasks $(T_i^1, \ldots, T_i^n)$ and the set of execution conditions $(EC_i^1, \ldots, EC_i^m)$. The vertices $\vartheta_i$ are a set of tasks configurations. The edges $\psi_i$ are a set of conditions to enable a reconfiguration of controllers as shown in Fig. 2.

The reconfiguration of intelligence is based on situation awareness and system execution context. For this, we propose that each rule has an execution condition to run a reconfiguration of controller as mentioned above and it has also a super activation/deactivation condition to reconfigure the rule base ERB at run-time. In this sense, let $AC_i = \{AC_i^1, \ldots, AC_i^h\}$ the triggers for adapting ERB to new environment requirements based on context-awareness.

$$AC_i \rightarrow \Gamma(t) \quad \text{Op} \quad \{\gamma_j, \gamma_j \in \Gamma\} \quad (8)$$

$$\text{Op} = + \quad | \quad - \quad (9)$$

$$AC_i = \{AC_1, \ldots, AC_h\} \quad (10)$$

$$\text{SMM} = (\beta, \phi) \quad (11)$$

SMM is the state machine to formalize the $\Gamma_i(t)$ behaviour in reconfigurable system. Indeed, the vertices $\beta$ are a set of rules in ERB $(\gamma_i^1, \ldots, \gamma_i^m)$ and $\phi$ are a set of activation conditions $(AC_i^1, \ldots, AC_i^h)$. The RIA module presented in the system architecture with Fig. 1, has the ability to reconfigure the rule base of the system by providing the flexibility shown in Fig. 2.

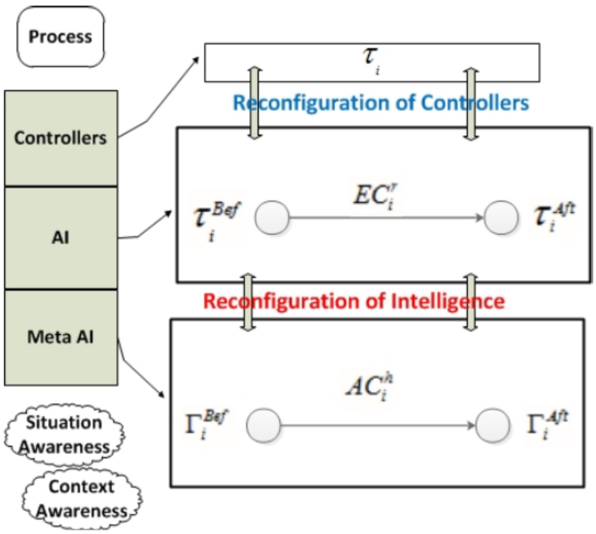


**FIGURE 2.** Levels of reconfiguration in DIS.

### C. INTELLIGENCE QoS FACTOR

The adaptive behaviour of the tasks and the rule base should be guaranteed by a QoS factor.

For this reason, we propose to evaluate the intelligence QoS of the system at run-time depicted as $IF_g(t)$ and calculated following this proposed formulas:

$$IF_i(t) = \frac{\sum_{j=1}^{n_i} IR(T_i^j, t)}{n_i(t) + m_i(t)} \tag{12}$$

$$IF_G(t) = \min\{\int IF(\Theta_i^j, t)dt, i \in [1, k]\} \tag{13}$$

where IR: $T_i \rightarrow \{0, 1\}$ is a boolean function to follow the status of each task $T_i^j$ if it is running then IR=1 else IR=0. Indeed, an intelligent system is an adaptive control system with compressed number of production rules. Behind this semantic, to decide whether the system should update its behaviour or its intelligence we should quantify the tasks and the rules in the current system configuration. Where the process of decision making is composed of rules and task's configurations, the intelligence QoS in DIS is presented as the minimum number of active decisions from all the running tasks and rules in the whole DIS. The proposed factor is null when all the running tasks haven't corresponding rules in ERB and it tends to 1 when all the running tasks have a corresponding rules in ERB and $m_i(t)$ is minimal (equal to 1). The cost of reconfiguration process for both intelligence $Cost_{RI}$ and controllers $Cost_{RC}$ in time computation is proportional to the intelligence QoS. To evaluate the overall cost of the proposed reconfiguration restriction, we propose the following formulas:

$$GeneralCost = \frac{Cost_{RI} + Cost_{RC}}{(N_i + m_i) - (n_i(t) + m_i(t))} * \frac{1}{IF_G} \tag{14}$$

The overall cost is exponential when $IF_G$ tends to zero and it is minimal when $IF_G$ tends to one.

$$Performance = \frac{1}{GeneralCost} \tag{15}$$

The performance of the proposed system architecture is proportional to the cost where it is null when the *GeneralCost* is exponential.

### D. PROBLEM AND CONSTRAINTS

In the literature, there are a lot of successful studies addressing the scheduling problem of real-time tasks [34], [44]. For example, the inherent flexibility of hierarchical structure scheme with main-servo loop control structure is proposed to the problem of integrated chassis control system for the vehicle. It includes both main loop which calculates and allocates the aim force using the optimal robust control algorithm and servo loop control systems which track and achieve the target force using the on board independent brake actuators [45]. The work in [42] proposes a stack resource policy (SRP) that allows processes with different priorities to share a single run-time stack. In [34] work, the authors suggest the rate monotonic policies (RM) and the earliest deadline first (EDF) for the scheduling of periodic tasks. The original priority ceiling protocol (OPCP) and immediate priority ceiling protocol (IPCP) are presented in [40] in order to manage the tasks that share resources. The goal of the priority ceiling protocol is to prevent deadlocks and reduce the blocking to at most one critical section [43]. The (m,k)-firm model [39] is used in a degraded mode to better characterize the timing constraints of real-time streams. Also, various related works have been dedicated to develop reconfigurable real-time control systems. Also, the research in [41] focuses on low-power dynamic reconfigurations of synchronous real-time control systems. On the other hand, the work in [9] proposes new solutions to schedule reconfigurable real-time systems implemented with independent periodic and probabilistic tasks under real-time constraints. However, the new generation of embedded systems aims to address new criteria such as flexibility and agility [17]. To reduce their cost, these systems should be changed and adapted to their environment without any disturbance by applying reconfiguration scenarios. We distinguish two reconfiguration policies [7], [8]: (i) static reconfigurations applied off-line to apply changes before system could start, (ii) dynamic reconfigurations applied dynamically at run-time. Two cases exist in the last policy: manual reconfigurations applied by users and automatic reconfigurations applied by intelligent agents. In the current work, we are interested in dynamic and automatic reconfigurations.

The reconfiguration process in our work is exactly an architectural contribution for autonomous behaviour and adaptive control in DIS. Nevertheless, updating tasks and rules configurations may broke the semantic of the system correctness where shared resources in local devices and cooperative decisions between different devices halt.

Shared resources $ShRes_i$ are local resources shared between different tasks running in the same device. Cooperative decisions are controlled by a set of exchanged messages *CoMsg* between different devices. We denote $CoMsg^{\tau}_{i,o,j} = (T_i^j, T_o^j, Data^{\tau}_{i,o,j})$ the message exchanged between $\Theta_i$ and $\Theta_o$. Those messages may contain shared data, precedence constraints or synchronized release time between two devices $\Theta_i$ and $\Theta_o$ using the parameter $Data^{\tau}_{i,o,j}$ for dependent tasks. For the distributed rules, we denote $CoMsg^{\Gamma}_{i,o,j} = (\gamma^j_i, F_o, Data^{\Gamma}_{i,o,j})$ the exchanged data between rules to activate/deactivate a set of rules or/and to update the values of the working memory of the RBS $F_o$ in the device $\Theta_o$. The problem that we invoke in this research paper is making the dynamic reconfiguration technology suitable and feasible for both tasks and rules in DIS. We denote $Cstr_{RT}$ the real-time feasibility, $Cstr_{Pow}$ the energy feasibility and $Cstr_{Intl}$ the intelligence quality of service feasibility as the constraints of the proposed architecture. The following formulas should be guaranteed over the system evolution and execution:

$$\frac{Cstr_{RT} * Cstr_{Pow}}{Cstr_{Intl}} \geq 1 \tag{16}$$

$$\begin{aligned} Cstr_{RT} &= 1 \quad \text{If } \sum_{j=1}^{N} \frac{C^j_i}{P^j_i} \\ &\preceq (\sum_{i=1}^{n} n * (2^{\frac{1}{n}} - 1)) - \text{X} \end{aligned} \tag{17}$$

$$Cstr_{Pow} = 1 \quad \text{If E} \preceq E_{Cr} \tag{18}$$

where E is the actual battery charge status. The real-time feasibility is calculated in function with the processor utilization for the RM scheduling algorithm [34]. If the conditions presented in equations 17 and 18 become non-satisfiable the values of their constraints change to 0.

## III. CONTRIBUTION

In this section, we present the proposed approach to resolve the problem invoked in this research paper for the coordination in AI-based adaptive DIS.

### A. MOTIVATION

In a distributed platform, contradictory reconfiguration process affects the semantic of the system correctness and feasibility. In this context, we propose a new protocol to guarantee the system constraints presented in equation 15 and to preserve reliability in each device and the coordination, synchronization and cooperation of the whole system. Indeed, the proposed protocol should assure reliability in both intelligence and controllers components by a dedicated coordination factor. This new useful protocol for coordination is making for distributed platform and system Quality of Service (QoS) in complex and robust system implementation, presents new solutions for guaranteeing the management of the intelligence QoS of the whole distributed discrete-event system, proposes new solutions for constructing dynamically the correctness of the coordination between the different decisions by the field of the coordination factor under inferring time and memory consumption constraints, and coordinates between the different devices in a distributed platform. We note that these contributions are relevant to any dynamic reconfiguration of intelligence for high behaviour adaptability of autonomous distributed intelligent systems that have the considered architecture under the considered constraints of this paper.

### B. FORMALIZATION OF THE PROPOSED PROTOCOL

First, we propose to formalize the protocol for shared resources in local devices then we explore the features of coordination intra-devices.

#### 1) COORDINATION IN LOCAL DEVICE

Let $Reconf_i$ denotes the reconfiguration data structure made of a *type*, *scenarios* and a *value*. The type of reconfiguration has "Local" or "Distributed" values, the scenarios are a controller reconfiguration $Reconf^{\tau}_i$ or/and a reconfiguration of intelligence $Reconf^{\Gamma}_i$. The value is formalized as follows:

$$\begin{aligned} Reconf_i.\text{Value} = \{&+\{\alpha^{+}_{\in\tau}\}o\{\beta^{+}_{\in\Gamma}\}; -\{\alpha^{-}_{\in\tau}\}o\{\beta^{-}_{\in\Gamma}\}; \\ &: \{\alpha^{:}_{\in\tau}\}o\{\beta^{:}_{\in\Gamma}\}\}. \end{aligned} \tag{19}$$

where, the operators $+$, $-$ and : present the operations of addition, delete and update parameters respectively and the sets $\alpha$ or $\beta$ may be null depending on the scenarios of reconfiguration (controllers or/and intelligence).

The shared resources in local devices are in two levels; AI and Meta-AI. Indeed, the set of rules in $\Gamma_i(t)$ may have contradiction with a reconfiguration scenarios $Reconf^{\Gamma}_i$ and the same thing for $\tau_i(t)$ with $Reconf^{\tau}_i$. We denote $ShRes^{\Gamma}_i$ as all the couple of contradictions in $\Gamma_i$ by the following statements:

$$ShRes^{\Gamma}_i : \Gamma_i \rightarrow (\Gamma_i)^* \tag{20}$$

$$ShRes^{\Gamma}_i(\gamma^j_i) = (\gamma^{j1}_i, \gamma^{j2}_i, \gamma^{j3}_i, \ldots) \tag{21}$$

Dealing with tasks, we propose $ShRes^{\tau}_i$ as the set of couple presenting inconsistency and contradictions when running in the same device.

$$ShRes^{\tau}_i : \tau_i \rightarrow (\tau_i)^* \tag{22}$$

$$ShRes^{\tau}_i(T^j_i) = (T^{j1}_i, T^{j2}_i, T^{j3}_i, \ldots) \tag{23}$$

#### 2) COORDINATION IN DISTRIBUTED RECONFIGURATION

The distributed reconfiguration is a process that should be managed with a coordination protocol to ensure the correctness of results between the devices. The feasibility of the initial system is done by a matrix $\text{CM} = \binom{M_\tau}{M_\Gamma}$ to save all the dependencies in both rules and tasks:

$$M_\tau = \begin{array}{c} \\ T_1 \\ T_2 \\ \vdots \\ T_r \end{array} \begin{array}{c} \begin{array}{cccc} \Theta_1 & \Theta_2 & \cdots & \Theta_k \end{array} \\ \begin{pmatrix} \delta^{\tau}_{1,1} & \delta^{\tau}_{1,2} & \cdots & \delta^{\tau}_{1,k} \\ \delta^{\tau}_{2,1} & \delta^{\tau}_{2,2} & \cdots & \delta^{\tau}_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \delta^{\tau}_{r,1} & \delta^{\tau}_{r,2} & \cdots & \delta^{\tau}_{r,k} \end{pmatrix} \end{array} \tag{24}$$

where $T_{j\in[1..r]}$ are all the tasks sharing resources in the whole DIS where r $\preceq \sum_{i=1}^{k} n_i$. The set $\delta^{\tau}_{j,i}$ is a collection of tasks running in the device $\Theta_i$ and having dependencies with the corresponding $T_j$ in the matrix (for independent tasks, $\delta^{\tau} = 0$

and they don't figure in CM).

$$M_\Gamma = \begin{array}{c} \\ \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_q \end{array} \begin{array}{c} \begin{array}{cccc} \Theta_1 & \Theta_2 & \cdots & \Theta_k \end{array} \\ \begin{pmatrix} \delta^\gamma_{1,1} & \delta^\gamma_{1,2} & \cdots & \delta^\gamma_{1,k} \\ \delta^\gamma_{2,1} & \delta^\gamma_{2,2} & \cdots & \delta^\gamma_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \delta^\gamma_{q,1} & \delta^\gamma_{q,2} & \cdots & \delta^\gamma_{q,k} \end{pmatrix} \end{array} \quad (25)$$

In the same way, $\gamma_{j \in [1..q]}$ are all the rules having shared decisions behaviour depicted in the matrix as $\delta^\gamma$ where q $\preceq \sum_{i=1}^{k} m_i$. We propose the equation of the CF as follows:

$$\epsilon^\tau = \sum_{i=1}^{k} \sum_{j=1}^{r} Card(\delta^\tau_{j,i}) \quad (26)$$

$$CF^\tau = \frac{\sum_{i=1}^{r} \sum_{j=1}^{k} (\sum_{v=1}^{Card(\delta^\tau_{j,i})} IR(T_v))}{\epsilon^\tau} \quad (27)$$

$$\epsilon^\Gamma = \sum_{i=1}^{k} \sum_{j=1}^{q} Card(\delta^\Gamma_{j,i}) \quad (28)$$

$$CF^\Gamma = \frac{\sum_{i=1}^{q} \sum_{j=1}^{k} (\sum_{v=1}^{Card(\delta^\Gamma_{j,i})} IR(\gamma_v))}{\epsilon^\Gamma} \quad (29)$$

$$CF = CF^\Gamma * CF^\tau \quad (30)$$

$$CF = \frac{\epsilon^\tau * \sum_{i=1}^{q} \sum_{j=1}^{k} (\sum_{v=1}^{Card(\delta^\Gamma_{j,i})} IR(\gamma_v))}{\epsilon^\Gamma * \sum_{i=1}^{r} \sum_{j=1}^{k} (\sum_{v=1}^{Card(\delta^\tau_{j,i})} IR(T_v))} \quad (31)$$

The coordination feasibility of the whole $\Sigma$ system is ensured by the CF equation where $\epsilon$ is the cardinality of all the $\delta^\tau$ subs. IR is a boolean function defined as follow IR: $\delta^\tau_{j,i} \cup \delta^\Gamma_{j,i} \Rightarrow \{0, 1\}$. For each task $T_{j=1..r}$ having dependency $\delta^\tau_{j,i}$, we investigate the status of each member of the collection $T^u_{j,i} \in \delta^\tau_{j,i}$; if it is currently running on its corresponding device $\Theta_i$ then IR($T^u_{j,i}$) = 1 else IR($T^u_{j,i}$) = 0. Also, we investigate the rule base to fetch whether the shared decisions depicted by $\delta^\Gamma$ are omnipresent then IR($\gamma^u_{j,i}$) = 1 else IR($\gamma^u_{j,i}$) = 0.

*Proposition 1:* The DIS has a correct behaviour when CF = 1 and intelligent QoS behaviour when $IF_g(t)$ tends to 1. In particular, each reconfiguration scenario $Reconf_i$ should guarantee the coordination feasibility by a primitive of CF denoted as $CF_{Reconf_i}$. To calculate this primitive, we propose a reconfiguration matrix of dependencies as follows:

$$CM_{Reconf_i} = \begin{pmatrix} M_\tau[Reconf_i.value.\alpha^1_{i,j}] \\ M_\tau[Reconf_i.value.\alpha^2_{i,j}] \\ \vdots \\ M_\Gamma[Reconf_i.value.\beta^1_{i,j}] \\ \vdots \\ M_\Gamma[Reconf_i.value.\beta^h_{i,j}] \end{pmatrix} \quad (32)$$

### C. IMPLEMENTATION

We propose three modules RI, RC and RBS for a real-time linux distribution; RI is implemented as an interrupt service routine where a vector of interrupts defined as the values of $AC_i$ handles the different intelligence reconfiguration scenarios. RC is implemented as a loadable kernel module [27] to create/delete or/and reconfigure tasks parameters at run-time. RBS is the module of decision making for adapting the tasks in function of environment changes based-on the rules residing in ERB. Finally, a dedicated module for local reconfiguration is implemented as a part of the proposed protocol CoRDIS where its algorithm is shown in Fig. 3. To deal with those contradictions, a set of reconfiguration priorities are provided for each current/new configuration of running rules and tasks described respectively as: $Pr^{i,Bef}_{Reconf,\Gamma}$ and $Pr^{i,Bef}_{Reconf,\tau}$ for current configurations $Pr^{i,Aft}_{Reconf,\Gamma}$ and $Pr^{i,Aft}_{Reconf,\tau}$ for the new coming reconfiguration scenarios. The reconfiguration process is denied when the running configuration of tasks/rules has higher priority than the coming contradictory reconfiguration process. Otherwise, the process of reconfiguration is executed after deleting the running contradictory tasks/rules. Those priorities are used for both local and distributed reconfiguration process.

```
Data: Reconf_i : A Reconfiguration Scenarios, τ_i(t) : All the running tasks,
      Γ_i(t) : All the running rules, ShRes_i^Γ : All the dependent rules,
      ShRes_i^τ : All the dependent tasks
Result: ReconfigurationStatus
begin
    if (Reconf_i.type = "Local") then
        if (Reconf_i^τ != ∅) then
            foreach T_i^j ∈ Reconf_i^τ do
                if T_i^j ∈ ShRes_i^τ then
                    if Pr_{Reconf,τ}^{i,Aft} < Pr_{Reconf,τ}^{i,Bef} then
                        ReconfigurationStatus = "Denied";
                        Exit(0);
                    else
                        Del_i^τ += T_i^z;
        if (Reconf_i^Γ != ∅) then
            foreach γ_i^j ∈ Reconf_i^Γ do
                if γ_i^j ∈ ShRes_i^Γ then
                    if Pr_{Reconf,Γ}^{i,Aft} < Pr_{Reconf,Γ}^{i,Bef} then
                        ReconfigurationStatus = "Denied";
                        Exit(0);
                    else
                        Del_i^Γ += γ_i^z
    ReconfigurationStatus = "Accepted";
    Run(Reonf_i, Del_i^Γ, Del_i^τ);
```

**FIGURE 3.** **Algorithm of local reconfiguration for CoRDIS.**

To face the disturbance caused by the distributed reconfiguration process, we propose the token-ring implementation of CoRDIS as a full solution for reconfigurable DIS with $IF_G(t)$ and CF guarantee. Indeed, two phases are proposed as an implementation of token-ring protocol; *Phase 1*: Asking the slave's devices to accept/update/deny the reconfiguration scenarios, *Phase 2*: Running/Cancelling the distributed reconfiguration proposed from a master device $\Theta_i$.

#### 1) PHASE 1

Prospecting the Slaves: In this phase, a negotiator token is implemented as a frame containing the address of the master, the data which is the reconfiguration scenarios and finally a dedicated two bits for accepting (01), denying (00) or updating (10) the reconfiguration scenarios. At first, the token is initialized from the master with the proposed reconfiguration scenarios and the bits of decisions have the value of 01. If a device denies the reconfiguration process caused by a contradiction as mentioned in local reconfiguration, the bits become 00 and the address of destination is reconfigured to be the address of the master. Indeed, the process of reconfiguration is denied when the priority of the coming reconfiguration scenarios is lower than the priority of the current local configuration of this slave. If a device updates the reconfiguration scenarios, the bits of decision become 10 and the successors devices have only the ability to deny the reconfiguration process by changing the bits to 00. In this case, a successor device that accept the reconfiguration process should only liberate the token to its successor without modifying the frame.

#### 2) PHASE 2

The Master Decision: After prospecting the devices, the master decides whether to accept/deny or renegotiate the devices. In case of accepting, the master sends an executive token containing the scenarios of reconfiguration to be executed. In case of updating, the master decides whether to renegotiate the devices by re-sending the negotiator token with the updated reconfiguration scenarios or to deny the reconfiguration process. To avoid the exponential complexity of the renegotiation process, we propose a soft deadline of negotiation process, $D_i^{Neg}$ for each master device as a value of response time starting from the first phase. The deadline of negotiation of each node is t. After each iteration, CoRDIS verifies if the response time is usually lower than $D_i^{Neg}$ and re-sends the negotiator token. When the response time halts the previous condition, the process of reconfiguration is finally denied. As a soft deadline, CoRDIS tolerates that the response time be higher than $D_i^{Neg}$ in the last iteration; i.e., if the iteration u verifies the condition and the iteration u + 1 does not but the bits of decisions are updated to 01 as accepting the reconfiguration process, in this case the negotiation process is closed with a satisfiable status and the phase 2 is executed and the executive token is send to the slave's devices. Otherwise, the reconfiguration process is cancelled when a device denies the reconfiguration process or the response time exceed the deadline of negotiation and the bits of decision still to 10. The whole implementation of CoRDIS for distributed reconfiguration is presented in the state chart diagram depicted in Fig. 4.

## IV. EXPERIMENTATION

The goal of this experimental part is to explain the different steps of the proposed methodology and to prove the efficiency of our original approach. First, we proceed to the implementation of a brief application of the system architecture and CoRDIS protocol. Later, we propose to evaluate our contribution with performance study for memory consumption, inference response time, intelligence QoS and system coordination feasibility.

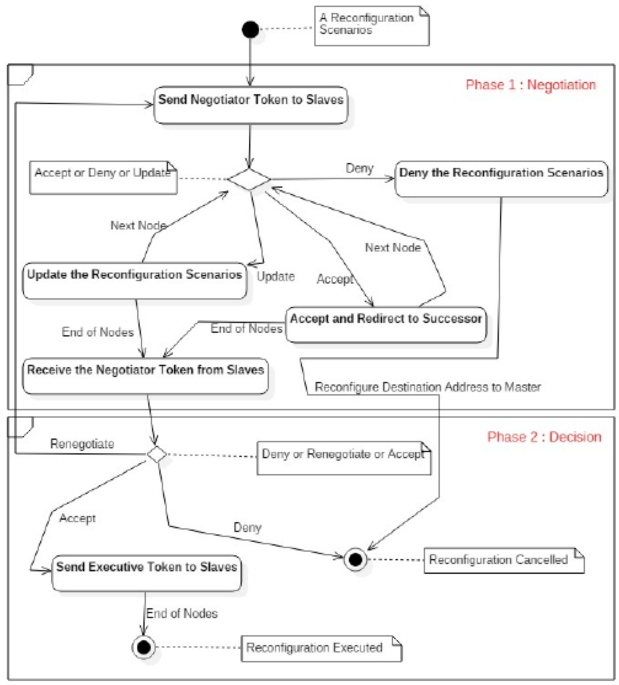


**FIGURE 4.** CoRDIS for distributed reconfiguration.

### A. APPLICATION

As an application of the proposed architecture, we present the following system description $\Sigma = \{\Theta_1, \Theta_2, \Theta_3\}$ as depicted in Fig. 5 where a set of rules to perform the reconfiguration of controllers and a set of tasks in each device. The intelligence reconfiguration process is depicted in Fig. 6 where for each device, we present a set of activation conditions to reconfigure the rule base.

The dependencies are presented as follows:

$$CM = \begin{array}{c} \\ T_1^1 \\ T_1^3 \\ T_2^1 \\ T_2^2 \\ T_2^5 \\ T_3^1 \\ \gamma_2^5 \\ \gamma_3^1 \end{array} \begin{array}{c} \begin{array}{ccc} \Theta_1 & \Theta_2 & \Theta_3 \end{array} \\ \left( \begin{array}{ccc} \cdots & T_2^2 & T_3^1 \\ \cdots & T_2^5 & \cdots \\ \cdots & \cdots & T_3^1 \\ T_1^1 & \cdots & \cdots \\ T_1^3 & \cdots & \cdots \\ T_1^1 & T_2^1 & \cdots \\ \cdots & \cdots & \gamma_3^1 \\ \gamma_1^2 & \gamma_2^5 & \cdots \end{array} \right) \end{array} \quad (33)$$

We suppose the following scenarios; first, the configuration of the system is as follows: $\{AC_1^1\ o\gamma_1^1, AC_2^2\ o\gamma_2^2, AC_3^2\ o\gamma_3^2\}$. An external event in $\Theta_2$ infers the rule-based system to run $\gamma_2^1$ as a reconfiguration of controller process. This later presents a "distributed" reconfiguration scenarios where $Reconf_2^{\tau}.value = \{-\{T_2^1, T_2^2\}; +\{T_2^5, T_2^6\}\}$. To guarantee the dependencies depicted in CM, the device $\Theta_1$ should delete $T_1^1$ and $T_1^2$ and adds $T_1^3$ and the device $\Theta_3$ should delete $T_3^1$. In this case, we suppose that $Pr_{Reconf,\tau}^{2,Aft}$ has the higher priority than $Pr_{Reconf,\Gamma}^{1,Bef}$ and $Pr_{Reconf,\Gamma}^{3,Bef}$. The master device $\Theta_2$ sends the negotiator token to $\Theta_3$ which updates the

| Devices | Rules | $\tau(t)$ |
|---|---|---|
| $\Theta_1$ | $\gamma_1^1$ | $T_1^1, T_1^2$ |
| | $\gamma_1^2$ | $T_1^1, T_1^4$ |
| | $\gamma_1^3$ | $T_1^1.P = P_{max}, T_1^2.P = P_{max}, T_1^3$ |
| $\Theta_2$ | $\gamma_2^1$ | $T_2^5, T_2^6$ |
| | $\gamma_2^2$ | $T_2^1, T_2^2$ |
| | $\gamma_2^3$ | $T_2^3, T_2^4, T_2^5$ |
| | $\gamma_2^4$ | $T_2^1, T_2^7, T_2^5$ |
| | $\gamma_2^5$ | $T_2^2, T_2^6, T_2^7$ |
| | $\gamma_2^6$ | $T_2^1, T_2^4$ |
| $\Theta_3$ | $\gamma_3^1$ | $T_3^2, T_3^3$ |
| | $\gamma_3^2$ | $T_3^1, T_3^3$ |
| | $\gamma_3^3$ | $T_3^2, T_3^4, T_3^5$ |

**FIGURE 5.** Adaptive control component.

| Devices | AC | $\Gamma(t)$ |
|---|---|---|
| $\Theta_1$ | $AC_1^1$ | $\gamma_1^1, \gamma_1^2$ |
| | $AC_1^2$ | $\gamma_1^1, \gamma_1^3$ |
| $\Theta_2$ | $AC_2^1$ | $\gamma_2^5, \gamma_2^6$ |
| | $AC_2^2$ | $\gamma_2^1, \gamma_2^2$ |
| | $AC_2^3$ | $\gamma_2^3, \gamma_2^4, \gamma_2^5$ |
| $\Theta_3$ | $AC_3^1$ | $\gamma_3^1$ |
| | $AC_3^2$ | $\gamma_3^2, \gamma_3^3$ |

**FIGURE 6.** Adaptive intelligence component.

reconfiguration scenarios to be $\{-\{T_3^1, T_1^1, T_2^1, T_2^2\}; +\{T_2^5, T_2^6\}\}$ and succeeds the token to $\Theta_1$ with the bits of decision 10. In the same way, $\Theta_1$ has less priority than the coming reconfiguration process then, it accepts the reconfiguration scenarios and sends the token to the master. $\Theta_2$ accepts the reconfiguration process and sends the executor token which contains the final reconfiguration process value and the $Reconf_2^{\tau}.value = \{-\{T_3^1, T_1^1, T_2^1, T_2^2\}; +\{T_2^5, T_2^6\}\}$ process of reconfiguration is executed.

### B. EVALUATION OF PERFORMANCE

In this section we propose to measure the system intelligence QoS, the coordination correctness, the execution time and the memory consumption and we will compare it to a static system $\Sigma_2$ without a reconfiguration features. In this simulation study, we propose a GMB having 10000 rules for 10000 tasks and we present an investigation about the memory allocation for the rule base and the inference time. The system $\Sigma_2$ has the GMB and all the tasks running in each device. The system $\Sigma_1$ has an initial configuration running of rules depicted in an ERB and a set of tasks with the proposed approach and modules. We have compared our solution to the static system $\Sigma_2$ and we get satisfiable results with an optimization on the system execution time by managing the set of tasks and rules in the system $\Sigma_1$. The solution gets good results by

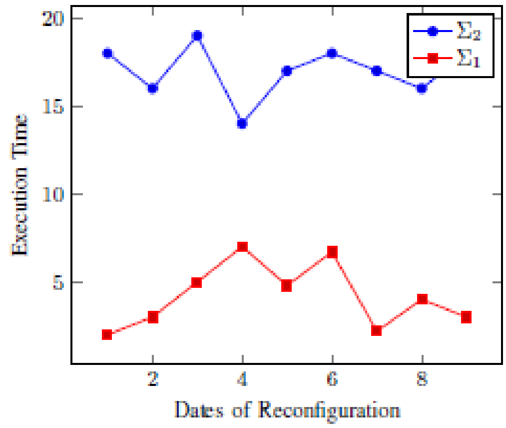


**FIGURE 7.** Execution time for $\Sigma_1$ and $\Sigma_2$.

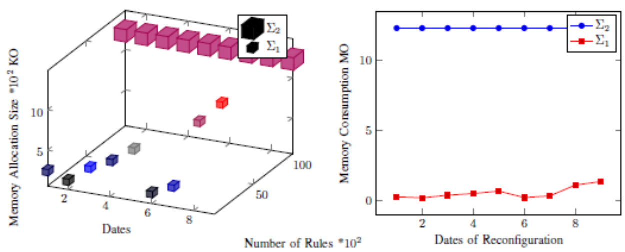


**FIGURE 8.** Memory consumption for $\Sigma_1$ and $\Sigma_2$.

minimizing the time inference and the whole system response time like it is shown in Fig. 7.

Our contribution is implemented in embedded system where low memory constraint enhanced the community to develop techniques to minimize the system allocation. In this sense, the simulation presented in Fig. 8 shows that by reconfiguring the rule base and the set of tasks the memory consumption is optimized where only the rules and the tasks for the current contextual description presented in the two components AI and Meta-AI are running in DIS. Dealing with intelligence QoS and coordination correctness, we present to evaluate our solution described by the system $\Sigma_1$ compared to $\Sigma_2$ and without the features of reconfiguration and coordination. The system $\Sigma_1$ preserves its intelligence QoS with a little disturbance in the moment of reconfiguration as depicted in Fig. 9. Also, we evaluate the coordination correctness with the field of CF variation as shown in Fig. 9, the system has a valid and a correct coordination behaviour when all running tasks/rules satisfy the CM matrix.

In order to be more effective, we can apply our study to many experimentations since our methodology covers various aspects: Coordination, inference response time feasibility and minimization of the memory consumption. We can for example, apply our approach for developing main-servo loop integrated chassis control system, the harmonically excited non-linear suspension system using a pair of symmetric viscoelastic buffers, dual axe drive pure electric vehicle based on motor loss model and big data calculation. And also for

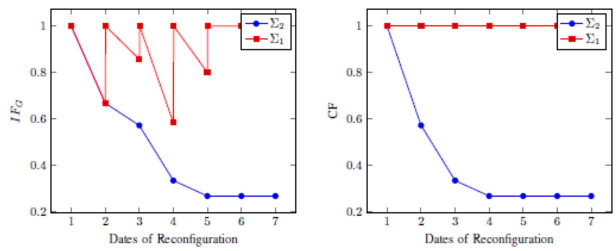


**FIGURE 9.** System coordination correctness and intelligence QoS status.

| *Work* | *Adaptive Control* | *Adaptive Intelligence* | *Coordination* | *Rule Base Partitioning* |
|---|---|---|---|---|
| [13] | Yes | No | Yes | Without Rule-based Implementation |
| [19] | No | Can be Implemented for Adaptive Rule-based System | No | Yes |
| [18] | No | Can be Implemented for Adaptive Rule-based System | No | Yes |
| [20] | A Platform for Adaptive Real-time DIS | Should be supported with frameworks integration | It has the ability for distributed coordination | Without Rule-Based Integration |
| [17] | Yes | No | No | No |
| Our Work | Yes | Yes | Yes | Yes, The activation conditions may be produced from a dedicated algorithm of rule partitioning or user defined activation conditions |

**FIGURE 10.** Comparative studies.

more IoT and cloud computing (CC) field applications under the considered constraints of this paper but we are restricted in this paper work to just apply it to a RTDroid as a real-time based Linux distribution and a dedicated RBS implementation with Drools for dynamic controllers reconfiguration as a case study for explain this methodology to manage distributed intelligence reconfiguration scenarios.

### *C. DISCUSSION*

A whole implementation of the proposed architecture is successfully tested with RTDroid [38] as a real-time based Linux distribution and a dedicated RBS implementation with Drools for dynamic controllers reconfiguration. The proposed architecture may be applied to a various domain of application for digital industry, autonomous car or unmanned vehicle, defence purpose and smart cities where adaptive control should be enhanced by expert domain implemented in a knowledge base. The activation conditions can be applied with user requirements or dedicated rule base partitioning as works in [36], [37], and [14]. By comparing our work with the existing methods above [12], [19], [33], [36], [37] we believe that our contribution is original since these related works do not consider the same assumptions of this work. To the author knowledge, this proposed approach and the coordination protocol for both adaptive intelligence and control is the first work in this field as a full implementable software architecture. We present in Fig. 10 a full description of the works related to our contribution with a description of their adaptive control, adaptive intelligence, coordination and finally rule base partitioning.

## V. CONCLUSION

In this research work we present a new solutions for adaptive DIS where we implement an adaptive rule-based system for both controllers and intelligence reconfiguration. A new protocol of communication for coordination in distributed and local reconfiguration called CoRDIS is proposed as the major contribution of this research work. Our methodology is certain since it can be applied to any system with the predefined assumptions. To the best of our knowledge, this paper is original since no one in all related works and in all our previous works deals with the automatic intelligence reconfiguration of reconfigurable distributed intelligent systems. The simulation and the experimentation have shown that by decreasing the number of rules in the rule base, the response time of the system becomes better and the memory consumption decreases. The correctness of the system in the coordination process and the intelligence QoS are ensured in this paper by proposing equations to verify the system behavior. In the future works, we aim to study the security part of the whole solution with cloud computing features and in internet of things (IoT) architecture. Also, we plan in a future work to deal with a real case study under other constraints such as energy and fault tolerance that should be satisfied by this kind of systems. Moreover, we will work on optimizing the application process of the proposed approach where we will consider the performance of this methodology.

## ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful and constructive comments that greatly contributed to improving the final version of the paper. They would also like to thank the Editors for their generous comments and support during the review process. Finally, they would like to thank everyone that made this research possible.

## REFERENCES

[1] O. Karoui, Z. Li, N. Wu, M. Khalgui, E. A. Nasr, and A. M. El-Tamimi, "One-step control-ahead approach for the design of an optimal petri-net based deadlock prevention policy," *IEEE Access*, vol. 6, pp. 34307–34323, 2018.

[2] I. Ghribi, R. B. Abdallah, M. Khalgui, Z. Li, K. Alnowibet, and M. Platzner, "R-Codesign: Codesign methodology for real-time reconfigurable embedded systems under energy constraints," *IEEE Access*, vol. 6, pp. 14078–14092, 2018.

[3] I. Khemaissia, O. Mosbahi, M. Khalgui, Z. Li, and T. Qu, "Coherence and feasibility of real-time software tasks in networked adaptive systems," *IEEE Access*, vol. 6, pp. 35824–35843, 2018.

[4] H. Chniter, Y. Li, M. Khalgui, A. Koubaa, Z. Li, and F. Jarray, "Multi-agent adaptive architecture for flexible distributed real-time systems," *IEEE Access*, vol. 6, pp. 23152–23171, 2018.

[5] W. Housseyni, O. Mosbahi, M. Khalgui, Z. Li, L. Yin, and M. Chetto, "Multiagent architecture for distributed adaptive scheduling of reconfigurable real-time tasks with energy harvesting constraints," *IEEE Access*, vol. 6, pp. 2068–2084, 2018.

[6] S. Theiss, V. Vasyutynskyy, and K. Kabitzsch, "Software agents in industry: A customized framework in theory and praxis," *IEEE Trans. Ind. Informat.*, vol. 5, no. 2, pp. 147–156, May 2009.

[7] H. Grichi, O. Mosbahi, M. Khalgui, and Z. Li, "New power-oriented methodology for dynamic resizing and mobility of reconfigurable wireless sensor networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 7, pp. 1120–1130, Jul. 2018.

[8] S. B. Meskina, N. Doggaz, M. Khalgui, and Z. Li, "Multi-agent framework for smart grids recovery," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1284–1300, Jul. 2017.

[9] X. Wang, I. Khemaissia, M. Khalgui, Z. Li, O. Mosbahi, and M. Zhou, "Dynamic low-power reconfiguration of real-time systems with periodic and probabilistic tasks," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 258–271, Jan. 2015.

[10] A. B. Ahmed, O. Mosbahi, M. Khalgui, and Z. Li, ''Toward a new methodology for an efficient test of reconfigurable hardware systems,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1864–1882, Oct. 2018.

[11] M. O. B. Salem, O. Mosbahi, M. Khalgui, Z. Jlalia, G. Frey, and M. Smida, ''Brometh: Methodology to design safe reconfigurable medical robotic systems,'' *Int.J. Med. Robot. Comput. Assist. Surg.*, vol. 13, no. 3, p. 1786, Sep. 2017.

[12] H. Grichi, O. Mosbahi, M. Khalgui, and Z. Li, ''RWiN: New methodology for the development of reconfigurable WSN,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 109–125, Jan. 2017.

[13] G. Pratl, D. Dietrich, G. P. Hancke, and W. T. Penzhorn, ''A new model for autonomous, networked control systems,'' *IEEE Trans Ind. Informat.*, vol. 3, no. 1, pp. 21–32, Feb. 2007.

[14] W. Lakhdhar, R. Mzid, M. Khalgui, Z. Li, G. Frey, and A. Al-Ahmari, ''Multiobjective optimization approach for a portable development of reconfigurable real-time systems: From specification to implementation,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 3, pp. 623–637, Mar. 2018.

[15] O. Karoui, M. Khalgui, A. Koubâa, Z. Li, E. Guerfala, and E. Tovard, ''Dual mode for vehicular platoon safety: Simulation and formal verification,'' *Inf. Sci.*, vol. 402, pp. 216–232, Sep. 2017.

[16] J. Zhang, M. Khalgui, Z. Li, G. Frey, O. Mosbahi, and H. Ben Salah, ''Reconfigurable coordination of distributed discrete event control systems,'' *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 323–330, Jan. 2015.

[17] M. Gasmi, O. Mosbahi, M. Khalgui, L. Gomes, and Z. W. Li, ''R-node: New pipelined approach for an effective reconfigurable wireless sensor node,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 6, pp. 892–905, Jun. 2018.

[18] W. Lakhdhar, R. Mzid, M. Khalgui, Z. Li, G. Frey, and A. Al-Ahmari, ''Multiobjective optimization approach for a portable development of reconfigurable real-time systems: From specification to implementation,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 3, pp. 623–637, Mar. 2019. doi: 10.1109/TSMC.2017.2781460.

[19] X. Wang, Z. Li, and W. Wonham, ''Dynamic multiple-period reconfiguration of real-time scheduling based on timed DES supervisory control,'' *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 101–111, Feb. 2016.

[20] M. Weiser, ''The computer for the 21st century,'' *SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 3, no. 3, pp. 3–11, Jul. 1999.

[21] R. Davis and J. J. King, ''Rule-Based Expert Systems,'' in *The Origisn of Rule-Based Systems in AI*. B. G. Buchanan and E. H. Shortliffe, Eds. AAAI Press, 1984.

[22] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2010, Ch. 1,16,14.

[23] V. J. Davidson, ''Expert systems in process control,'' *Food Res. Int.*, vol. 27, no. 2, pp. 121–128, 1994.

[24] G. J. Nalepa and P. Ziecik, ''Integrated embedded prolog platform for rule-based control systems,'' in *Proc. Int. Conf. Mixed Design Integr. Circuits Syst., (MIXDES)*, Gdynia, Poland, Jun. 2006, pp. 716–721.

[25] B. J. Koo, Y. S. Park, and S. Yang, ''Microcontroller implementation of rulebased inference system for smart home,'' *Inter. J. Smart Home*, vol. 8, no. 6, pp. 197–204, 2014.

[26] J. Davis, J. Hoffert, and E. Vanlandingham, ''A taxonomy of artificial intelligence approaches for adaptive distributed real-time embedded systems,'' in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, Grand Forks, ND, USA, May 2016, pp. 233–238.

[27] I. Khemaissia, O. Mosbahi, M. Khalgui, and W. Bouzayen, ''New reconfigurable middleware for feasible adaptive RT-linux,'' in *Proc. 4th Int. Conf. Pervasive Embedded Comput. Commun. Syst. (PECCS)*, Jan. 2014, pp. 158–167.

[28] B. Kaimal, A. R. Metkar, and G. Rakesh, ''Self learning real time expert system,'' *Int. J. Soft Comput., Artif. Intell. Appl.*, vol. 3, no. 2, pp. 13–25, May 2014.

[29] N. Buduma, *Fundamentals of Deep Learning*. Sebastopol, CA, USA: O'Reilly Media, 2015.

[30] G. Estrin, ''Reconfigurable computer origins: The UCLA fixed-plus-variable (F+V) structure computer,'' *IEEE Ann. Hist. Comput.*, vol. 24, no. 4, pp. 3–9, Oct./Nov. 2002.

[31] M. Khalgui and O. Mosbahi, ''Intelligent distributed control systems,'' *Inf. Softw. Technol.*, vol. 52, no. 12, pp. 1259–1271, Dec. 2010.

[32] W. Bouzayen, H. Gharsellaoui, and M. Khalgui, ''New solutions for intelligence adaptation of reconfigurable embedded systems,'' in *Proc. 30th Eur. Simulation Modeling Conf. (ESM)*, Las Palmas, Spain, Oct. 2016, pp. 233–242.

[33] W. Bouzayen, H. Gharsellaoui, and M. Khalgui, ''New solutions for AI-based adaptive system under real-time and low-memory constraints,'' in *Proc. 23rd Int. Conf. Parallel Distrib. Process. Techn. Appl. (PDPTA)*, Las Vegas, NV, USA, pp. 16–22, Jul./Oct. 2017.

[34] C. L. Liu and J. W. Layland, ''Scheduling algorithms for multiprogramming in a hard-real-time environment,'' *J. ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973.

[35] E. E. Almeida, J. E. Luntz, and D. M. Tilbury, ''Event-condition-action systems for reconfigurable logic control,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 2, pp. 167–181, Apr. 2007.

[36] H. Rui and L. Chunmei, ''A large knowledge base partitioning and reasoning study based on graph theory,'' in *Proc. 6th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)* Tianjin, China, Aug. 2009, pp. 203–207.

[37] L. Ook, ''The automatic clustering of a rule base for its maintenance,'' in *Proc. Amer. Conf. Inf. Syst. (AMCIS)*, Aug. 1997, pp. 265–270.

[38] Y. Yan, K. Dantu, S. Y. Ko, J. Vitek, and L. Ziarek, ''Making android run on time,'' in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp. (RTAS)*, Pittsburgh, PA, USA, Apr. 2017, pp. 25–36.

[39] M. Hamdaoui and P. Ramanathan, ''A dynamic priority assignment technique for streams with (m, k)-firm deadlines,'' *IEEE Trans. Comput.*, vol. 44, no. 12, pp. 1443–1451, Dec. 1995.

[40] L. Sha, R. Rajkmumar, and J. P. Lehoczky, ''Priority inheritance protocols: An approach to real-time synchronization,''İ*IEEE Trans. Comput.*, vol. 39, no. 9, pp. 1175–1185, Sep. 1990.

[41] F. Yang, N. Wu, Y. Qiao, M. Zhou, and Z. Li, ''Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 3, pp. 502–516, Mar. 2017.

[42] A. Burns, M. Gutiérrez, M. A. Rivas, and M. González Harbour, ''A deadline-floor inheritance protocol for EDF scheduled embedded real-time systems with resource sharing,'' *IEEE Trans. Comput.*, vol. 64, no. 5, pp. 1241–1253, May 2015.

[43] M. Uzam, Z. Li, G. Gelen, and R. S. Zakariyya, ''A divide-and-conquer-method for the synthesis of liveness enforcing supervisors for flexible manufacturing systems,'' *J. Intell. Manuf.*, vol. 27, no. 5, pp. 1111–1129, Oct. 2016.

[44] Y. Hou, N. Wu, M. Zhou, and Z. Li, ''Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm,'' *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 3, pp. 517–530, Mar. 2017.

[45] R. Zhang, Z. He, H. Wang, F. You, and K. Li, ''Study on self-tuning tyre friction control for developing main-servo loop integrated chassis control system,'' *IEEE Access*, vol. 5, pp. 6649–6660, 2017.

[46] M. G. Abidi, M. B. Smida, M. Khalgui, Z. Li, and T. Qu, ''Source resizing and improved power distribution for high available island microgrid: A case study on a tunisian petroleum platform,'' *IEEE Access*, vol. 7, pp. 22856–22871, 2018.

[47] R. Oueslati, O. Mosbahi, M. Khalgui, Z. Li, and T. Qu, ''Combining semi-formal and formal methods for the development of distributed reconfigurable control systems,'' *IEEE Access*, vol. 6, pp. 70426–70443, 2018.

[48] R. Idriss, A. Loukil, M. Khalgui, Z. Li, and A. Al-Ahmari, ''Filtering and intrusion detection approach for secured reconfigurable mobile systems,'' *J. Elect. Eng. Technol.*, vol. 12, no. 5, pp. 2051–2066, Sep. 2017.

**HAMZA GHARSELLAOUI** received the B.S. degree in computer science from Tunis El-Manar University, Tunis, Tunisia, in 2004, and the M.S. degree in industrial computer science from THE National Institute of Applied Sciences and Technology (INSAT), Carthage University, Tunis, in 2007, where he did his research in computer science with the National Institute of Applied Sciences and Technology, to receive the Ph.D. degree, in 2013. He was a Researcher in computer science with the Khurmah University College, Taif University, Saudi Arabia. He was a Temporary Lecturer with UBO University, Brest, France. He is currently a part-time Researcher and an Assistant Professor with the National Engineering School of Carthage (ENICarthage), Carthage University. He is active in several projects and in other interesting international collaborations.

**MOHAMED KHALGUI** received the B.S. degree in computer science from Tunis El Manar University, Tunis, Tunisia, in 2001, the M.S. degree in telecommunication and services from Henri Poincaré University, Nancy, France, in 2003, the Ph.D. degree from the National Polytechnic Institute of Lorraine, Nancy, in 2007, and the Habilitation Diploma degree in information technology (computer science) from the Martin Luther University of Halle-Wittenberg, Halle, Germany, in 2012, with Humboldt Grant. He was a Researcher in computer science with the Institut National de Recherche en Informatique et Automatique, Rocquencourt, France; the ITIA-CNR Institute, Vigevano, Italy; the Systems Control Laboratory, Xidian University, Xi'an, China; and the KACST Institute, Riyadh, Saudi Arabia; a Collaborator with the SEG Research Group, Patras University, Patras, Greece; the Director of the RECS Project, O3NEIDA, Canada; the Director of the RES Project, Synesis Consortium, Lomazzo, Italy; the Manager of the Cyna-RCS Project, Cynapsys Consortium, France; and the Director of the BROS and RWiN Projects, ARDIA Corporation, Germany. He is currently a Professor with Jinan University, China. He has been involved in various international projects and collaborations. He is a TPC member of many conferences and different boards of journals.