Journal of Intelligent & Robotic Systems (2024) 110:112
https://doi.org/10.1007/s10846-024-02148-6

REGULAR PAPER

Effects of the Human Presence among Robots in the ARIAC 2023 Industrial Automation Competition

Leandro Buss Becker[1,2] · Anthony Downs[3] · Craig Schlenoff[3] · Justin Albrecht[3] · Zeid Kootbally[3] · Angelo Ferrando[4] · Rafael Cardoso[5] · Michael Fisher[1]

Received: 5 December 2023 / Accepted: 11 July 2024

Abstract

ARIAC is a robotic simulation competition promoted by NIST annually since 2017, aiming to present competitors' with contemporary industry problems to be solved using agile robotics. For the 2023 competition, ARIAC competitors must perform assembly and kitting tasks by controlling four autonomous ground vehicles (AGVs), one floor-based robot, and one ceiling-based (Gantry) robot in an attempt to overcome a range of agility challenges in the supplied simulated environment, itself based on the Robot Operating System (ROS 2) and Gazebo. The 2023 competition also included a "human" agility challenge, comprising a (simulated) human operator working among robots on the factory floor. This development was motivated by the fact that, while robots and automation play an increasingly significant role in modern manufacturing, there still remains a close relationship between machines and humans. They should complement each other's strengths and cover each other's limitations while also observing any required safety rules. For example, the ISO standard "Robots and Robotic Devices – Collaborative robots" (ISO 15066:2016) prescribes the distances required between humans and robots. Within the ARIAC simulation environment, each human operator is controlled using autonomous Belief-Desire-Intention (BDI) agents. At the same time, competitors can monitor the position of each human operator at any time by subscribing to the relevant ROS topic. In this article, we analyse the effects of this (simulated) human presence in the 2023 ARIAC competition and perform a detailed analysis of how the three different human personalities that were implemented affect the assembly tasks undertaken at the four different locations of the assembly stations. Given how the system is currently implemented, it appears that the influence of each encoded personality on the competitors is not as predictable as anticipated. We expand on why this may be a problem when addressing real collaborative spaces involving humans and industrial robots and the improvements that can be undertaken to mitigate the ensuing problems.

Keywords Human-robot collaboration · Intelligent systems · BDI agents · Robot motion planning

1 Introduction

The US National Institute of Standards and Technology (NIST) has been organising the "Agile Robotics for Industrial Automation Competition" (ARIAC)[1] annually since 2017 [1]. The ARIAC competition brings together researchers and industry professionals to address challenges in agile robotics that the industrial sector is now encountering. The primary goal of ARIAC is to provide real-life manufacturing scenarios in which humans and robots collaborate in a low-volume, high-mix workload in a shared environment. As highlighted in [2], the synergy of skills of both robots and humans will be essential in the smart factories of the future. For example, humans can carry out regular inspection visits to look for subtle defects or inconsistencies that our machines might miss. They can also reach every spot of the work-cell relatively quickly to intervene, for instance, in case of malfunction or unexpected behaviour. Humans can also perform periodic maintenance and occasional repairs and, not least, a human presence can also benefit the wider workforce's morale.

[1] https://www.nist.gov/ariac

Anthony Downs, Craig Schlenoff, Justin Albrecht, Zeid Kootbally, Angelo Ferrando, Rafael Cardoso and Michael Fisher contributed equally to this work

✉ Leandro Buss Becker
leandro.bussbecker@manchester.ac.uk

Extended author information available on the last page of the article

Published online: 31 July 2024

The importance of human involvement in automation has led to a new human-related challenge being introduced in the ARIAC 2023 competition. This challenge involves a (simulated) human operator moving along the factory floor to supervise four workstations. In addition to controlling a ceiling robot[2] to perform the assembly tasks, competitors must also, for safety reasons, avoid close contact between such a Gantry robot and the human operator patrolling the factory floor. The Gantry robot is required not to get any closer to the human operator than the distances established in the ISO 15066:2016 standard ("Robots and robotic devices – Collaborative robots"), which addresses safety issues around robot speed and separation [3]. Following these safety rules is paramount, as having humans and robots share a collaborative space can be dangerous. Recently, in South Korea, a robotics company employee was killed by a robot while performing an inspection procedure [4]. The same news item states that, some months before, another worker was injured by a robot in an automobile manufacturing plant. Consequently, in ARIAC 2023, competitors' are penalised if these ISO restrictions are violated.

ARIAC's environment is based on the Robot Operating System (ROS 2[3]) [5], an open-source framework that provides a wide variety of libraries and tools for creating robot software, complemented by Gazebo[4] [6], a 3D physics-based simulator. The combination of ROS-2 and Gazebo provides a versatile and practical platform for the development, testing, and refinement of robotics applications[5].

1.1 Representing Humans via Cognitive Agents

As described by Wooldridge [7], an agent is an abstraction developed in order to capture autonomous behaviour in complex and dynamic systems. An agent is also defined by Russell and Norvig [8] as an entity that "can be viewed as perceiving its environment through sensors and acting upon that environment through effectors". In agent architectures for autonomous systems, decision-making is encapsulated as a component programmed as an agent within a more extensive system. Since decision-making is encapsulated within these components then they are required to be, as much as possible, *rational agents*. These are loosely described as agents that will try to "do the right thing" [8].

Since the 1980s, this agent approach, and the concept of rational agents in particular, has spawned a vast range of research [9–14], not only regarding the philosophy behind autonomous decision-making but also around programming frameworks and practical industrial exploitation. Among these, the Beliefs-Desires-Intentions (BDI) model of agency [15], inspired by Bratman's theory of human practical reasoning [16], has emerged as the predominant mechanism for implementing rational decision-making. (Agents following this approach are alternatively referred to as cognitive agents.) As highlighted in [17], using rational agents with deliberative capabilities allows us to reduce development time, create programs with reduced descriptive complexity, and capture some basic "human-like" behaviours. A further aspect that motivates this use of BDI agents is that they ensure both transparency and verifiability (and, therefore, high levels of explainability and trustworthiness) [18].

Following the principles described in [19], we here assume the use of a hybrid agent architecture, with a BDI agent simulating the high-level decisions of the simulated human. More specifically, the implementation of the "human-like" agent was provided in Jason [20], a well-known BDI programming language [21]. (The details of our BDI agent implementation of humans in ARIAC was initially presented in [22].) Even though the simulated human always undertook the same inspection task, we designed the agent so that it could assume different behaviour types – from now on we refer to these types as "personalities". Our aim in having different personalities is to allow varying levels of interference to the Gantry caused by the human operator. Three different personalities were developed, termed: *helpful*, *indifferent*, and *antagonistic*.

1.2 Paper Contributions and Outline

This paper is devoted to discussing the results of the 2023 ARIAC competition and how the human agent presence impacts competitor behaviour. More specifically, examine evidence highlighting whether the different human personalities have the expected impact. We analysed two trials executed by ARIAC 2023 finalist teams and also conducted a more detailed analysis around how the three different human personalities affected assembly tasks conducted at the different locations (workstations) on the factory floor. Our results allow critical reflection on how humans and robots should coexist and collaborate in a shared space. They also open new perspectives on how the simulation scenario for future ARIAC editions could be improved.

A further contribution from this paper involves providing additional details about the ROS 2 components developed to allow the use of BDI agents within ARIAC 2023. This should

[2] Ceiling robots are also referred to as Cartesian or Linear robots. When the horizontal member is supported at both ends, as in the current work, they may also be called 'Gantry' robots. This terminology is provided by the ARIAC 2023 documentation and used throughout this paper.

[3] https://github.com/ros2

[4] http://gazebosim.org/

[5] Certain commercial products or company names are identified here to describe our competition. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the products or names identified are necessarily the best available for the purpose.

be of broader interest not only for future ARIAC competitors, but also for research groups that reuse our simulation scenario for their own research.

The paper is organised as follows. Section 2 describes the ARIAC competition, including the human agility challenge, and the different personalities that were implemented for the simulated human operator. Section 3 presents the software architecture for the overall ARIAC simulation environment, explaining how we integrate the synthetic humans in to it. The cognitive agent developed for controlling the human operator is further detailed in Section 4. We evaluate the influence of the different human personalities on the competitor scores in ARIAC 2023 in Section 5. Finally, we present our conclusions in Section 6.

2 The ARIAC Competition

As indicated above, ARIAC is an annual simulation-based competition that seeks to overcome the challenges currently confronting the industry around agile robotics. ARIAC's primary objective is to evaluate the "intelligence" of industrial robotic systems using agility metrics, ultimately enhancing both the autonomy and the productivity of these robots once deployed in manufacturing environments. This reduces the need for human workers to spend excess time on the, potentially hazardous, factory floor. Competition participants must develop control systems for both a floor robot and a ceiling robot (also termed a "Gantry robot"). The floor robot is used for 'kitting' tasks, while the Gantry robot supports both 'assembly' and kitting tasks.

Assembly is a manufacturing process during which interchangeable parts are (usually sequentially) added to an artefact to create the "end product". In ARIAC, assembly is simplified by allowing competitors to place parts in any order. For an assembly task, competitors are expected to use parts located on an AGV and to assemble those parts at one of the four "assembly stations". Kitting is a manufacturing process that groups together separate but related parts as one unit. For a kitting task, ARIAC competitors are expected to (a) place a kit tray onto one of the four AGVs, (b) place parts onto that kit tray in a specific quadrant, (c) direct the AGV to the warehouse, and (d) evaluate the submitted kit for scoring.

The ARIAC 2023 simulation and control environment uses Gazebo and ROS 2. The elements of this environment required to enable competitors to execute it and to develop and test their robotic control solutions are freely available for download[6]. The simulation environment (the Gazebo world) in which the ARIAC 2023 competition occurs is shown in Fig. 1. This image shows the Gantry robot performing an assembly task at assembly station AS #1, with support from AGV-2.

Previous to designing ARIAC in 2017, NIST analyzed other robotics competitions to ensure that they needed to further address industrial robotic agility. An example of analyzed competition was the Amazon Picking Challenge [23], which assessed the capability of robots to perform some of the everyday pick-and-place operations that humans currently perform. The Robot Perception Challenge [24] was another competition relevant to our agility challenges; the objective of this competition was to stimulate advancements in sensing and perception technologies for the next generation of robots. ARIAC was explicitly created to assess and evaluate the realm of Industrial Robot Agility comprehensively, as no other competitions effectively tackled this particular niche.

Agility challenges are broadly framed in ARIAC to encompass a range of issues and considerations, such as (i) task failure identification and robot recovery, (ii) automated planning to reduce (or eliminate) the time required for robot re-programming when introducing a new task, and (iii) operation in fluid environments, where robots can sense their environment and perform tasks anywhere in the workshop floor. ARIAC 2023 has eight "agility challenges"[7], each representing additional difficulties that competitors may face when performing kitting tasks. For example, competitors may encounter defective or inverted parts that may need to be discarded instead of used for assembly. The challenges are combined in a range of trials or competition runs, which competitors must successfully navigate in both the qualification and the final stages of the competition. In this paper, our particular emphasis is on the "human operator" agility challenge, that specifically examines the interactions between the Gantry robot and the human operator, as described below.

2.1 Human Operator Agility Challenge

In this challenge, a simulated human operator is introduced into the workcell. The main objective of this challenge is to assess the competitors' gantry robot control system and specifically its capacity for preventing collisions with the human operator. If the competitor team fails in this, a penalty results. (The human operator can be seen on the left-hand side of Fig. 1, while the gantry robot is located on the right-hand side.)

The simulated human operator assumes one of the three personalities in a trial. Once a personality has been selected for a trial, it remains constant throughout that trial. Although it is technically feasible to develop and implement dynamic personality changes during ARIAC, for the sake of simplifying evaluation, a decision was taken to use a static personality for the agent.

[6] https://github.com/usnistgov/ARIAC

[7] https://ariac.readthedocs.io/en/latest/competition/challenges.html

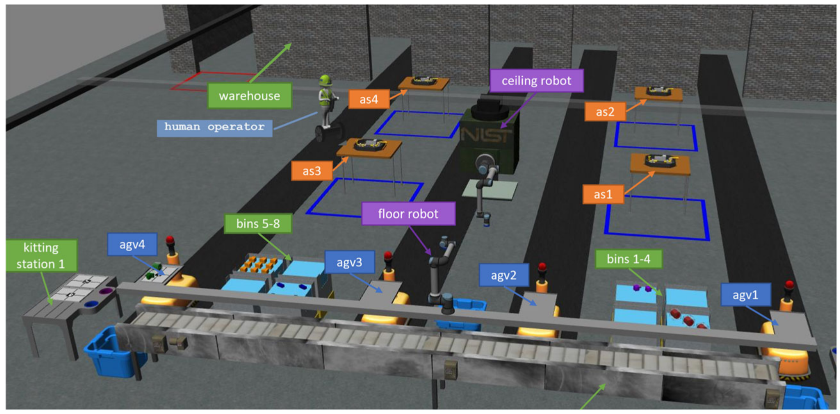


Fig. 1 A visualization of the ARIAC 2023 scenario. Here, the top left red square is the human operator initial position (also representing a "safe zone"). The Assembly Stations to be visited (numbered AS #1-4 in the paper) are the blue squares below the tables

Regardless of the personality adopted, the agent is programmed to avoid random movements and instead follows straightforward, pre-defined movement patterns across four workstations; this movement is meant to simulate everyday working and inspection tasks typically performed by humans in a factory setting. The human operator agent will continuously travel to these workstations and perform tasks until the trial completes.

In case the human operator and any of the robots come within a minimum safety distance of each other (calculation details are provided in the next section), the human operator is instantaneously transported (teleported) to a "safe zone", specifically the top-left position marked by a red rectangle in Fig. 1. The human operator is not transported if it gets close to a static AGV. In case the teleport operation is caused by being too close to the Gantry robot, the competitor team is penalised, and the Gantry robot is disabled for 15 seconds. After this period, regular operation is resumed. In these cases, the human operator is transported away purely to provide time for the competitors to recover, avoiding scenarios where the human could behave aggressively and force the Gantry robot into a deadlock.
The three agent personalities start with a plan to follow a predetermined path. However, this plan can be modified, as described below, providing a range of behaviours from non-intrusive to very intrusive:

1. *helpful* – Once the Gantry robot is within the human's line-of-sight[8], a *helpful* human operator will turn around, i.e., change its movement direction from clockwise to counter-clockwise or vice-versa.
2. *indifferent* – An *indifferent* human operator will always follow its predetermined path and directions, regardless of the location of the Gantry robot in its environment.
3. *antagonistic* – Once the Gantry robot is within the human's line-of-sight, an *antagonistic* agent will purposefully move towards the Gantry robot, aiming to interfere with the robot's current task.

The idea behind the *helpful* agent was that it behave in a non-intrusive manner, so that it would rarely interfere with robots. At the other extreme, the *antagonistic* agent was intended to be very intrusive and is likely to cause the most difficulties (and penalties) for competitors'. Given the more neutral characteristics of the indifferent agent, we expect this to be the one that best exposes the competitors' skills in avoiding contact with the human operator.

2.2 Calculation of Safety Distance

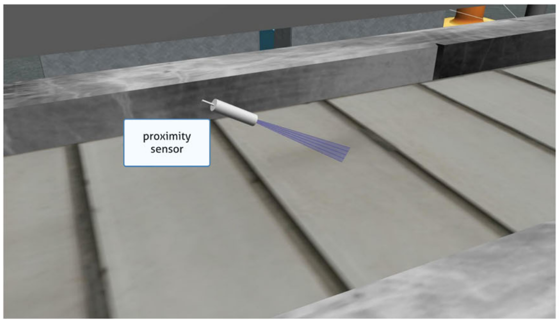
The "safety distance" between the human operator and any active robot (which state is *on*) is derived from the ISO/TS 15066:2016 standard ("Robots and robotic devices - Collaborative robots") that addresses safe robot speed and separation monitoring [3]. ISO/TS 15066:2016 specifies that the minimum allowable distance between a human and an active robot should be

$$d_{min} = k_H(t_1 + t_2) + k_R t_1 + B + \delta$$

where

- t_1 is the maximum time between actuation of the sensing function and the signal switching robotic devices to their *off* state,
- t_2 is the maximum response time of the robot (i.e., the time required to stop the robot),
- δ is an additional distance, based on the expected intrusion toward the critical zone before actuation of any protective equipment,
- k_H is the speed of the intruding human,
- k_R is the speed of the robot, and
- B is the Euclidean distance required to bring the robot to a safe, controlled stop.

[8] The line-of-sight is a parameter, and here we fix it as being (`safety distance` × 2).



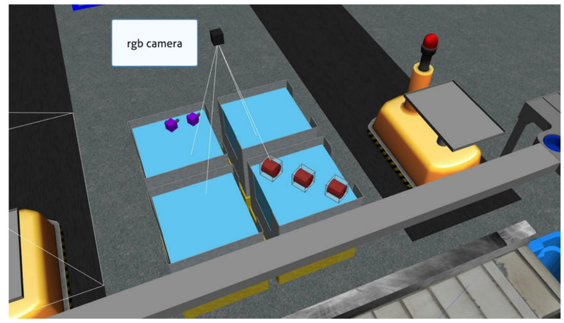


Fig. 2 Examples of ARIAC 2023 sensors: proximity sensor (left) and RGB camera (right)

It is important to highlight that for ARIAC 2023, the robots we are concerned about for the safety distance calculation are the Gantry and the four AGVs.

3 Gazebo Simulation Environment and ROS 2 Control Software

As previously stated, ARIAC 2023 requires two main software tools: the Robot Operating System (ROS 2) and the Gazebo simulation environment. They are typically used in conjunction for simulating robotic applications.

Gazebo is used to represent the virtual world where ARIAC takes place (depicted in Figs. 1 and 2). It contains both passive and active graphical elements as well as simulated sensors. The passive elements are the static ones, those graphical objects that do not move. On the other hand, the active elements represent those objects that can move and perform some action, such as the AGVs, the conveyor belt, the Gantry, and the human operator. Several sensors and cameras support monitoring both passive and active objects along the simulation. Competitors can select which sensors they wish to use, but each sensor has an associated cost that will be deducted from their final score. Figure 2 illustrates two typical sensors, a proximity sensor (on the left) and a RGB camera (on the right). The proximity sensor outputs how far an object is from the sensor and the RGB camera provides a RGB image. There are also more complex camera types available, including some giving depth information and even a camera that can provide a list of kit tray poses and part poses. The list of all available sensors is given in ARIAC documentation[9]. The information about the human operator (position and velocity) is, however, given for free to competitors. All competitors can easily access this information by subscribing to the appropriate ROS topic.

The control software is developed using ROS 2 components and represents the core element of ARIAC. It is in charge of most functionalities the simulation requires and manages the competition. For instance, it launches the simulation and the ARIAC Manager (AM) interface. The AM is used by competitors to interact with the simulation. The Competitor Control System (CCS) is the software that is provided by the competitors and that is responsible for communicating with the AM and executing the required tasks. Figure 3 illustrates the interactions among the CCS and the AM, which should be started using different Linux terminals.

In the first terminal, the ROS 2/Gazebo simulation environment is started with the AM. The latter manages the communications between the CCS and the ARIAC software. The competition's status undergoes several changes throughout a trial, and it is permanently published to the ROS-topic `/ariac/competition_state`. To implement the programming logic correctly, the CCS must subscribe to this topic.

As the trial begins, competitors should initiate the CCS from a second terminal. The initial task of the CCS is to launch the competition using the `/ariac/start_competition` service. It is important to note that the competition's status must be in the `READY` state before this service can be invoked. This service call initiates various actions, including the activation of robot controllers, sensor activation, commencement of the conveyor belt (if it is part of the trial), and the initiation of global challenges (if applicable in the trial). Instructions for kitting tasks will be communicated via the ROS-topic `/ariac/orders`. The outcome of this service call will transition the competition's status to `STARTED`.

After introducing new tasks, the CCS strives to fulfil and present orders using the robots. Order announcements can be based on timing, part placements, or the submission of previous orders. The CCS must use the `/ariac/submit_order` service to submit these orders, providing the order's ID as an argument.

Upon successfully submitting all orders, the CCS calls service `/ariac/end_competition`. This action transitions the competition's status to `ENDED`. Subsequently, the AM (ARIAC Manager) calculates the trial's scoring, con-

[9] Sensors: https://ariac.readthedocs.io/en/latest/competition/sensors.html

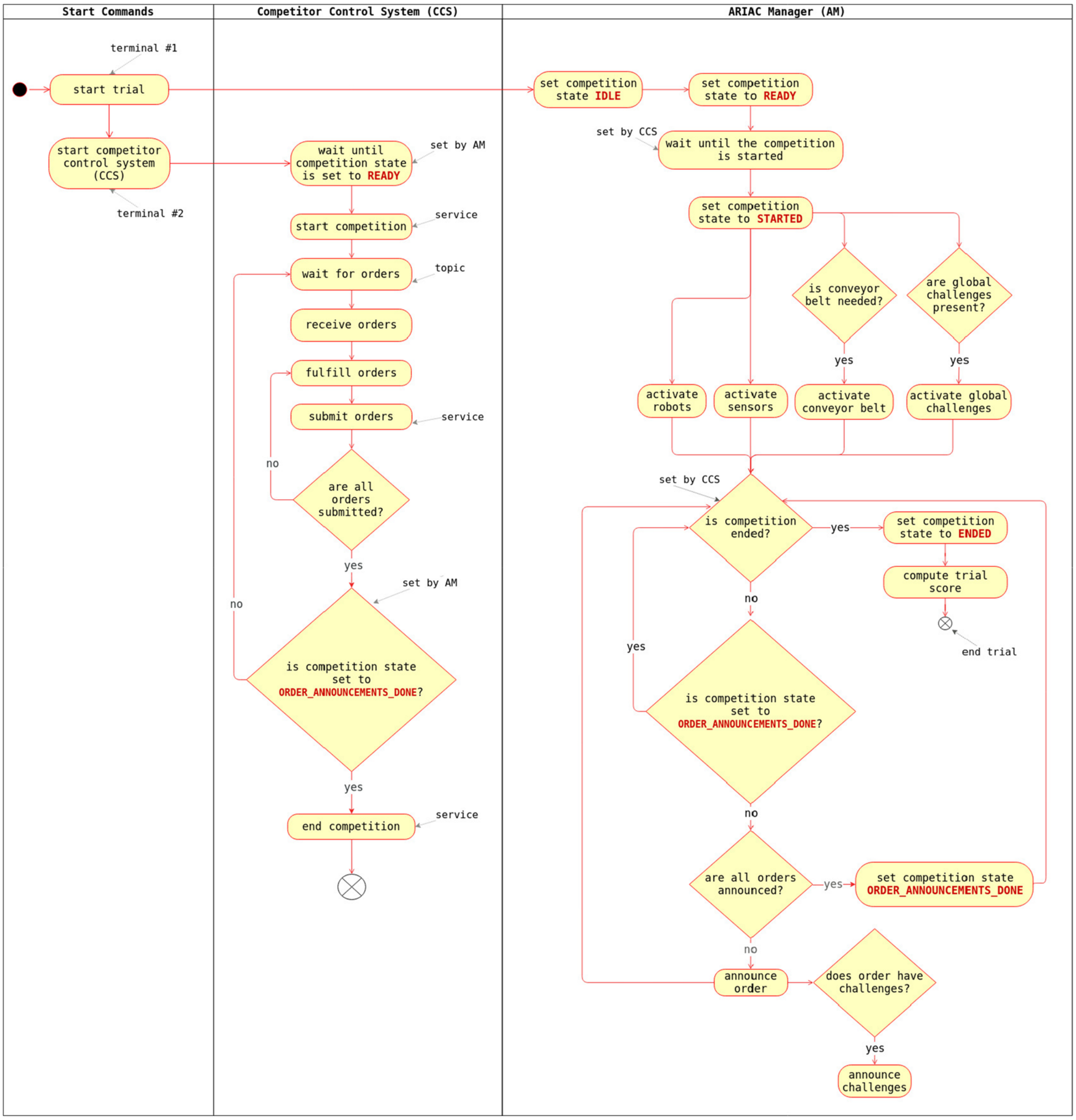


Fig. 3 ARIAC 2023 flowchart showing the interactions between the Competitor Control System (CCS) and the ARIAC Manager (AM)

cludes the trial, and stores the results. Before invoking the service to end the competition, the CCS must ensure that all orders have been announced. Finally, the competition's status is adjusted to `ORDER_ANNOUNCEMENTS_DONE` once all orders for the trial have been announced.

Presenting the ROS graph diagram – depicting the program's nodes, topics, and their interconnections – generated during ARIAC 2023 execution is impossible because of the high volume of nodes. There are circa 90 ROS nodes running while executing a trial. Performance tests conducted in [22] using a 24 cores Intel i9-10920X workstation show that 27 Linux processes are related to ROS 2/Gazebo. These processes required an average CPU utilisation of 510% (five cores entirely plus 10% of a sixth core).

3.1 Support for the Human Operator

In [22] we described all the ROS 2 elements (nodes, topics, services, actions, plugins) developed to support adding the human operator into the ARIAC 2023 environment. In the present paper, we restrict the discussion to the most important ones. A diagram containing a simplified view of such elements is presented in Fig. 4.

The *task_manager* Gazebo plugin, on the left side of the diagram, relates to the previously presented Ariac Manager (AM). It was coded in the CPP programming language. Among other things, it publishes the */ariac/start_human* ROS topic, which serves to start a Java process containing our (Jason) *human–agent*, detailed in the next section. Such

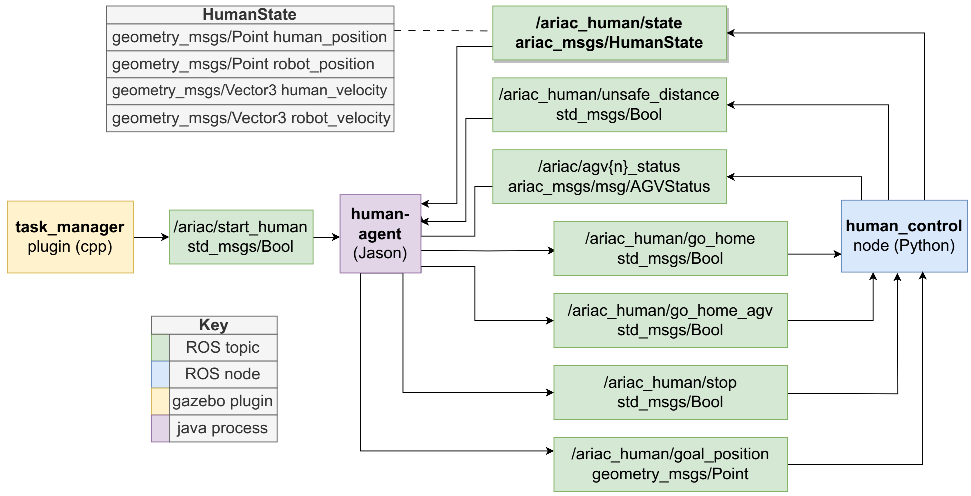


Fig. 4 Main ROS elements in ARIAC 2023 devoted for supporting the *human–agent*. Arrows represent the flow of information between different components

an agent can publish the four ROS topics on its right side (bottom) and subscribe to the three ROS topics also on its right (top). The topics that it subscribes to come from the *human_control* node. The *human_control* was programmed in Python and served to perform two main tasks: (i) navigate the human through the workcell; and (ii) teleport the human to the safe zone whenever necessary.

One last but important observation is that only one of these topics was explicitly informed to the competitors in the ARIAC 2023 documentation: */ariac/ariac_human/state*. By subscribing to this topic, competitors could identify the human's position and velocity. The output from such a topic is presented in Listing 1 for illustration purposes.

Listing 1 Output example from topic */ariac/ariac_human/state*

```
human_position:
  x: -14.99392<NL> y: -9.9999866<
     NL> z: 0.01002
robot_position:
  x: -7.000000<NL> y:  8.445e-08<
     NL> z: 0.70000
human_velocity:
  x: 5.658e-05<NL> y: -1.167e-06<
     NL> z: 2.8776e-05
robot_velocity:
  x: -9.607e-10<NL> y: 1.325e-10<
     NL> z: 0.0
```

4 Cognitive Agent Controlling the Human Operator

This section provides an overview of our *human–agent*, which, as previously mentioned, is responsible for controlling (taking the movement's decision) of the human operator. Readers interested in more details about the agent implementation and performance should refer to [22].

As introduced in Section 2.1, the human operator's primary job is making inspections in the four workstations present at the workcell. From a broad perspective, what the human operator does is to move along four predefined waypoints within the virtual factory's shop floor. The human will always start moving in a clockwise basis, starting at workstation 4 (AS #4) and continuing in the following order: "4 > 2 > 1 > 3 > 4 > ...". Since this is the first edition of ARIAC that incorporated the presence of humans with an impact in the score of the competitors, the human operator behaviour should not incorporate complex, random movements, aiming for predictability. The goal here was that ARIAC 2023 competitors' would not need very complex strategies to avoid close contact with the human operator.

The elements within the ARIAC 2023 scenario that interest the BDI agent are the following: the human operator, the four AGVs, and the Gantry. The relevant movement data (speed and location) about these elements are periodically updated within the agent's "memory" (belief base) through subscribing to the respective ROS topics. All ARIAC-related actions performed by the BDI agent are also sent to the *human_control* ROS node employing ROS topics. All these ROS topics are depicted in Fig. 4.

The behaviour of the human operator is illustrated in the flowcharts shown in Fig. 5. The flowchart on the left relates to the human operator with the indifferent personality, but it is a common aspect among the three different personalities. This can be observed by comparing it with the flowchart on the right, containing the antagonistic personality. The main task in both cases is to follow the predefined trajectory regardless of the position of the Gantry (left-side loop). Whenever the human operator is not within a safe distance of the Gantry, it is teleported to the safe zone (right-side loop). The difference in the antagonistic personality is that the human should start moving towards the Gantry when it sees it at a certain distance. The helpful personality is quite similar. The only difference is that when the Gantry is seen, then the human should move in the opposite direction of the Gantry.

4.1 The Jason Code

Agents programmed in Jason consist of: initial beliefs and rules; initial goals; and plans. Plans are written using the AgentSpeak(L) syntax [25] `triggering_event : context <- body.` wherein the `triggering_event` can be the addition/deletion of a `belief` or a `goal`, the `context` represents the preconditions of the plan, and the `body` relates with the sequence of operations (`actions` or addition/deletion of `beliefs/goals`). For the remainder of this section, we describe the most relevant plans in Jason for controlling the human.[10]

The main responsibility of the agent is to let human making the inspection procedures, that is, to let it move through the predefined waypoints. Therefore two different plans are used, with triggering events `+!work` and `+work_completed`, as presented in Listing 2. This code is, in fact, the same for the three agents' personalities. The `+work_completed` event is triggered when the human reaches its final position, fact that is reported by the *human_control* ROS node. There is one precondition for the related plan to be executed: the `counterClockWise` belief must be `TRUE`. The elements remaining within this plan's context, `working(Loc)& next_loc(Loc, Next)`, are used by the agent to determine which location should be visited next. A similar version of this plan exists for the case of the human performing a counterclockwise movement.

Listing 2 Plans related to visiting the workstations

```
+!work(Loc) : location(Loc, X,
 Y, Z) <-
    -working(_);
    +working(Loc);
    move(X, Y, Z).

+work_completed(_)
: working(Loc) & next_loc(Loc,
   Next) & counterClockWise
<- !work(Next).
```

The plan on Listing 3 represents the right-side loop of the previous flowcharts. The triggering event `+gantry_disabled(_)` happens when the gantry and the human operator are not at a safe distance. In this case, the agent must drop all its desires (ln.2) and also stop all the goals being executed, like for example moving to the target workstation. Then, it calls the *teleport_safe* external action, allowing to "teleport" the human to the predefined safe location (ln.3). Finally, the plan is concluded by invoking the `!work` plan in order to restore the default movement of the human operator by requesting it to the initial workstation. The '`!!`' operator is the addition of a goal that will be created as a separate intention stack as opposed to using a single '`!`' which would generate an intention in the same intention stack as the plan that added the goal.

Listing 3 Plan for when the Gantry and the human operator are not at a safe distance

```
1 +gantry_disabled(_) :
     firstStation(ST) <-
2     .drop_all_desires;
3     teleport_safe; // stop +
         teleport to safe zone
4     .wait(8000);
5     !!work(ST).
```

The code portion related with the *antagonistic* personality (yellow part on the right flowchart in Fig. 5) is shown in Listing 4. Its context clause serves to identify the current human destination (ln.2). Regarding the plan itself, it first stops and aborts any navigation goal (ln.3), then it leaves behind all desires (ln.4) and triggers the action that requests the human to start moving in direction of the Gantry (ln.5). Towards the end the plan it keeps the agent blocked waiting it to reach its target target position (ln.6). As this becomes true, it then starts moving to the next station (ln.7).

Listing 4 *Antagonistic* personality: Plan for when the human-operator and the Gantry get within sight-of-vision triggering a new behaviour in the human

```
1 +gantry_detected(_) :
2         working(Loc) & next_loc(
             Loc,Next) <-
3     stop_movement;
4     .drop_all_desires;
5     move_to_gantry;
6     .wait("+work_completed(_)");
7     !!work(Next).
```

5 Evaluation

In the first part of this section, we present the results related with the ARIAC 2023 human challenge. Afterwards, we per-

[10] The complete code can be found in the ARIAC's official repository 2023 release: https://github.com/usnistgov/ARIAC/releases/tag/v1.5-2023

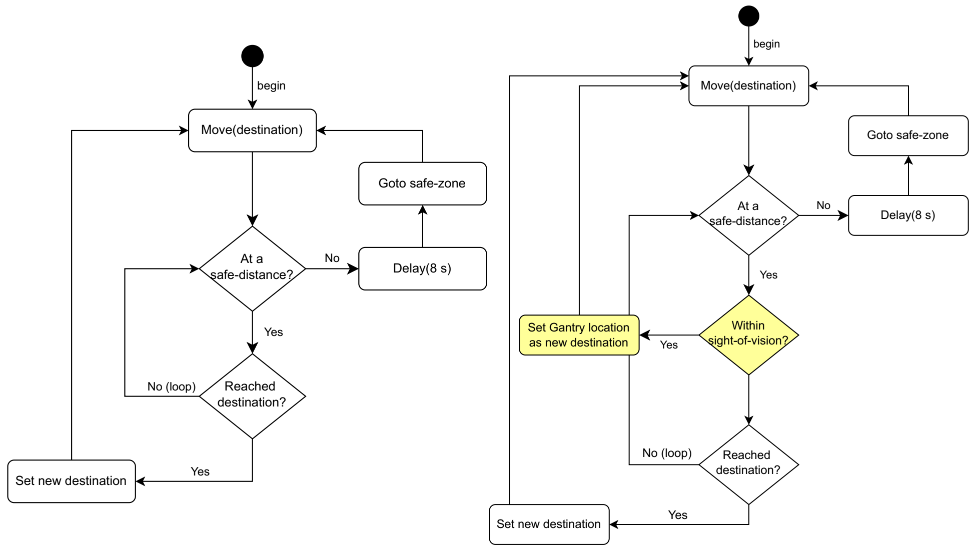


Fig. 5 Behaviour of two distinct human operator personalities: indifferent (left) and antagonistic (right). While the former always follow the predefined trajectory, the latter might change its movement pattern

form a more general evaluation concerning how the three different conceived human "personalities" influence the best scoring team of ARIAC 2023. We conclude this section by presenting some lessons learned and a discussion on future steps.

5.1 Results from the ARIAC 2023 Human Challenge

ARIAC 2023 finals consisted of ten different challenges (trials), named `final1...final10`. The last two, `final9` and `final10` are related with the human challenge. They consist of two distinct assembly tasks (adding interchangeable parts to a product). The Competitor Control System (CCS) can place parts and insert them in any order. For a trial where assembly tasks are required, the ARIAC environment starts with parts already placed on AGVs. The CCS is expected to:

1. Lock the AGV trays.
2. Move the AGVs to the correct assembly station.
3. Assemble the parts into a kit.
4. Submit the assembly task for scoring.

Both `final9` and `final10` trials consist of one assembly task announced at the beginning of the competition. The `final9` trial requires AGVs #1&2 to transport parts, so that assembly can occur at AS #1. The `final10` trial only requires parts from AGV #1 and assembly occurs at AS #2. Trial `final9` uses the *antagonistic* personality and trial `final10` uses the *indifferent* personality. Recalling Section 4, while the antagonistic human will try to get as close as possible to the Gantry, the indifferent human will follow a scripted path which may interfere with the Gantry's path.

Only two teams managed to submit the assembly trials, which for anonymity purposes we will name *Team #1* and *Team #2*. While the *Team #1* successfully finished both trials, the *Team #2* did not successfully complete either of them. The reason for this failure is not related with the humans but with the solution's inability to properly handle the parts to be assembled.

The results collected from executing these two trials are presented in Table 1. The first column represents the time needed to complete the trial, with α representing infinite time (it does not complete). The second column represents the quantity of penalties (Qtt-P) received by the Gantry due to the proximity to the human. For the *Team #2*, in the first trial the simulation crashed even before the first penalty. In the second trial, it crashed shortly after the fourth penalty, in the very last action before completing the assembly task, after about 120 s of simulation time.

Analysing the simulation videos, it is clear that none of the teams had developed any strategy for the Gantry to avoid contact with the human operator. Therefore, the following questions emerged:

Table 1 Performance of the teams in the two humans-related trials in ARIAC 2023

	`final9`: (AS #1) - *Antagonistic*		`final10`: (AS #2) - *Indifferent*	
	Time(s)	Qtt-C	Time(s)	Qtt-C
Team #1	60.4	1	76.8	2
Team #2	α	0	α	4

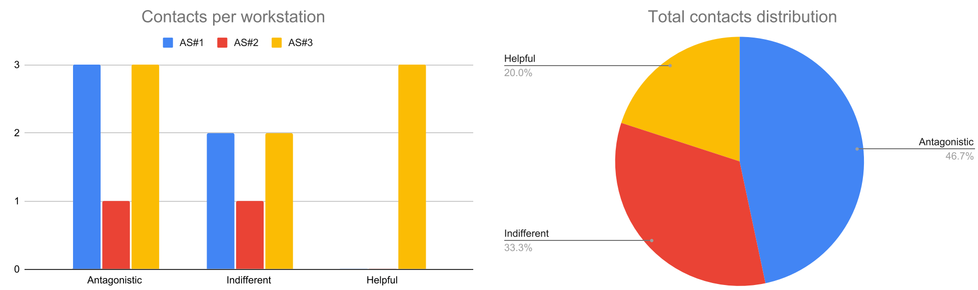


Fig. 6 Contact from the human personalities on the different trials: left chart shows the number of contacts per station and the right chart shows the total contacts distribution

- Do the *antagonistic* and the *indifferent* humans personalities always affect the robots similarly?
- Does the *helper* personality always prevent contact with the robot?

To explore these questions, we created and executed a new set of trials, as detailed in the next section.

5.2 Evaluating the Effects of the Three Different Human Personalities

To properly evaluate the effects of the three different human personalities and to be able to answer the questions previously presented, a new set of trials was orchestrated and performed after the competition. These trials were executed for *Team #1*, the only team that could complete the trials when the human operator was present. We used the same assembly task from trial `final10` at all four assembly stations (AS #1-4). Each new test trial was executed four times, one time for each of the three different human personalities and a fourth time without human presence. This fourth execution served as a time baseline, representing the fastest possible time for the team to complete the task without a human presence.

We analyse the results from the three human personalities on the first three stations (AS #1-3). The assembly station AS #4 is left out because the system entered a deadlock situation when it was used. This was because the human presence, regardless of the personality, caused too much contact with the robots. This kept the robot disabled for most of the simulation, and consequently, the assembly tasks could not be completed. This problem occurred due to the proximity of AS #4 to the safe zone. The charts in Fig. 6 depict the results related to the number of contacts. Table 2 summarises all obtained results.

Our first remark from the results obtained comes from the fact that performing assembly at the odd-numbered stations is faster than at the even-numbered ones – see the faster times from the "no-human" experiments in AS #1&3 vs. AS #2&4. This occurs because in AS #1&3 the AGV travels for a smaller distance to get into position, allowing the Gantry to start earlier. Next we discuss each of the trials presented in Table 2.

The first trial (AS #2) is basically the same as trial `final10`, therefore the results from the *indifferent* human execution are almost the same as those presented in the right columns from Table 1. These results reflect exactly what was expected for the three human personalities: the antagonistic is the one that causes more interference to the robot, which requires more time to complete the experiment, followed by the indifferent and finally by the helper, which did not cause any disturbance to the Gantry – it finishes its assembly job in the same time as if there were no humans present. Another interesting aspect that can be observed in this trial is that it highlights the more aggressive behaviour from the antagonistic human when moving towards the Gantry in comparison with the indifferent human. Even though, in this case, both antagonistic and indifferent humans follow the same path, the former approaches the Gantry faster than the latter and, as a consequence, causes more contacts (3 instead of 2).

Moving to the second trial (AS #1), the first observation is that the antagonistic human finishes the assembly slightly faster when compared with trial `final9`, where the Gantry is working at the same station. The reason here is that taking all four parts from the same AGV – which is the case in this trial – makes things faster than if taking the parts from

Table 2 Results from *Team #1* performing assembly at the four different stations and considering all possible human-operator configurations

Team	AS #2		AS #1		AS #3		AS #4	
	Time(s)	Qtt-C	Time(s)	Qtt-C	Time(s)	Qtt-C	Time(s)	Qtt-C
Antagonistic	92.7	3	57.6	1	86.8	3	α	α
Indifferent	77.4	2	56.1	1	70.9	2	α	α
Helpful	45.2	0	42.1	0	85.2	3	α	α
No-human[1]	45.3	-	41.8	-	40.9	-	47.4	-

[1]No human-operator was present in this trial

two different AGVs, such as it was done in trial `final9`. A surprising result from this trial was the fact that the indifferent human, using the path AS #4–2–1, caused the same level of interference as the antagonistic one, which was following the path AS #4–1. The indifferent human got too close to the Gantry when approaching AS #2. Although such behaviour is in accordance with our original specification, it leads to the following question: should the Gantry be penalised in case the human is transposing an area that is beyond the safety distance, but where the Gantry is not programmed to go to?

Analysing the results from AS #3 trial it is possible to observe the most unexpected result. The helpful human caused as many approximations to the Gantry as the antagonistic did. This is explained by the fact that no matter what is the path selected by the human, either AS #4–2 or AS #4–3, both will cause approximation. The path AS #4–2 will be similar to the previously provided explanation for the indifferent human in trial AS #1, leading to the same question.

Finally, in regards to the last trial (AS #4), the team was unable to successfully complete the task when there is a human present in the workcell. This comes from the fact that, by default, the human initially moves to AS #4. Consequently, the agent often comes into contact with the Gantry, causing the Gantry to be blocked on every attempt. Therefore the Gantry is unable to complete the assembly task. We decided to repeat this experiment changing the initial station for the human to visit, so that it starts the visits at AS #3. These results are shown in Table 3.

The numbers in Table 3 revealed a situation similar to the one present in the experiment AS #3, where the *helpful* agent performed worse (caused more approximations) than the *indifferent* one. The only difference here is that the *helpful* agent did not perform as bad as the *antagonistic* agent. Regardless, this is still an interesting result, as the helpful agent was designed to avoid contact with the Gantry.

By the end of these experiments we are able to answer the questions presented earlier, as follows.

1. Do the antagonistic and the indifferent humans personalities always affect the robots in the same way?
   Rpl.1: *Not always. The antagonistic agent normally causes more contact, as observed in 3 out of 5 experiments (60%). There were 2 experiments where they were even (caused the same number of contacts). But there was never a case where the indifferent agent caused more contacts than the antagonistic one.*
2. Does the helper personality always prevent contact with the robot?
   Rpl.2: *Not always. The helpful agent had contact with the Gantry in 3 out of 5 experiments. Clearly, having contact in 60% of the experiments means that the current strategy for the helpful agent must be modified if the intention for this personality is trying to avoid contact with the Gantry.*

Table 3 Modified AS #4 experiment: assembly task is still performed at station AS #4 but now the human operator is initially moving to station AS #3

	Modified AS #4	
	Time(s)	Qtt-C
Antagonistic	105.0	4
Indifferent	60.9	1
Helpful	74.0	2

5.3 Learned Lessons

The first learned lesson is about the fact that it proved to be difficult to properly characterise how the different human personalities will perform in respect to the amount of contacts with the Gantry that they will cause. This comes from the fact that the current planning for human movement ignores the task that is given to the Gantry. This would be improved if, at least, the human is conscious about the workstation where the Gantry is currently working on. Recalling the two different AS #4 experiments performed, if the human was aware that the Gantry would work at AS #4, then it should not head straight to AS #4 – which causes a deadlock – but instead would go to AS #3, as in the modified to AS #3 experiment.

Moreover, concerning whether the Gantry should be penalised in case the human is transposing an area that is beyond the safety distance, but where the Gantry is not programmed to go to, our view is that the answer should be *not always*. Of course we put safety as the top priority, so neither the Gantry nor any other robot should be operating beyond their safety distance to the human. But what if the robot stops, i.e., enters a standby mode? Then we suggest that it would be better to have no penalty applied to the Gantry. This would also open space for competitors to perform a "self-stop" in the Gantry, just like the "freeze" action that is currently performed by the CCS. However, such a self-stop would last less time then the current 15 s penalty time.

Finally, another aspect worthy of reflection is the fact that, currently, the human operator does not expose its intentions to the robots by means of interactions protocols (although the human's path was let clear to ARIAC 2023 competitors). Our view is that this would be an interesting action to be taken by the humans, because this would create conditions for the system planner to better schedule the robots' movements and ultimately cause fewer unplanned stops. As highlighted by Tausch and Kluge in [26], communication is important for humans and robots to share tasks and decide who will be in charge of task allocation.

5.4 Harmonious Coexistence in Future Human-Robot Interactions

In the experiments conducted we incorporated a single human operator in charge of inspection tasks. But what if we have more humans on the scene, in charge of different tasks? Let us suppose an additional human operator in charge of performing periodic maintenance procedures, and a further one is responsible for handling unexpected situations. How do we expect the robots to behave when these humans come into action? Do the humans need to approach their destination as-soon-as-possible (ASAP), or they could wait for a while?

We start with the case of an unexpected situation. When this happens, the goal is that it should be handled ASAP. This means that the human operator in charge should not lose a second and move towards the desired destination, as the indifferent agent does. For this to happen, the robots should "clear the path", so that the human can promptly reach the desired destination. Communicating with the robots about the situation and, more importantly, communicating the path that the human intend to take would give extra time for the robots to stop what they are doing and to position itself adequately. For instance, this would be really important if a global planning system is adopted, like in [27] where multi-agents are used.

Considering the human in charge of periodic maintenance, it does not really need to rush towards the destination. As the maintenance procedure is normally something that takes a relatively long time, then if the human looses a couple seconds waiting for a robot to finish its task it should not cause a large impact. This human would not behave like the any of the three personalities we have at the moment, we would require a different type of personality to represent such human behaviour. In any case, communicating the path that the human would take to the robot and vice-versa could be helpful in this situation. Alternatively, having the robot informing a "clear-to-go" message could also be helpful.

6 Conclusions

This paper was mainly intended to analyse the effects of the human presence in ARIAC 2023. Two competitors' teams managed to perform the trials related with the human agility challenge. Unfortunately, neither of these teams developed any kind of strategy to avoid close contact between the human operator and the Gantry. And this happened considering the fact that competitor's did not need to allocate any special sensor to find the human's location, it was available as a ROS topic.

Analysing this fact from the perspective of the winning team, our suspicion was that the penalty for such close contact was too "soft" and not worth it for the team to implement contact-avoidance strategies. In this respect, the competitors handled such a situation more like they were in a game environment than if they were in a real automated factory, with life-risky threats involved. While in a game they would win the "points" even without avoiding close contact, in a real factory, the violation of this safety rule is unacceptable. Our first conclusion is that more severe penalties should be applied in future ARIAC editions for trials that include human operators, such as attributing a score of zero to the trial in case of close contact between humans and active robots.

From the perspective of the different personalities that we crafted, it was possible to observe that the *helpful* personality did not behave as expected. While it was designed to be minimally intrusive, depending on the workstation that the Gantry had to work on the human was just as intrusive as the *indifferent* personality. And there was a situation where the *helpful* agent was as intrusive as the *antagonistic* agent. As discussed in the paper, changing such misleading behaviour involves several modifications, including the agent, the simulation scenario, and the simulation control.

A simple adjustment that could be adopted for future ARIAC editions is expanding the workcell, giving more free space between the workstations. Thereby the problems observed in experiments AS # 3-4-5 would likely not occur. As discussed in the previous section, defining communication procedures from both humans and robots might also be beneficial.

Finally, it should be clear that real collaborative scenarios should involve different human personalities, doing various tasks. For this to work in an optimised manner, it is essential to establish proper interaction protocols and resilient planning procedures in human-robot collaboration. This should also be taken into account in future ARIAC editions.

Acknowledgements The authors thank all NIST employees and interns involved in running ARIAC 2023 and, most importantly, to the teams that took part in the competition.

Author Contributions All authors contributed to the design of the so-called human agility-challenge. Angelo Ferrando, Leandro Buss Becker, Michael Fisher and Rafael Cardoso designed the human-operator BDI agent. Anthony Downs, Craig Schlenoff, Justin Albrecht and Zeid Kootbally were in charge of designing ARIAC 2023 simulation and conducting the competition. Material preparation was performed by Anthony Downs, Justin Albrecht and Zeid Kootbally. Data collection and analysis were performed by Leandro Buss Becker. The first draft of the manuscript was written by Leandro Buss Becker and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This research was supported by the National Institute of Standards and Technology (NIST), the University of Manchester and by the Royal Academy of Engineering Chair in Emerging Technologies programme.

Code Availability ARIAC 2023 source code, including the human-operator BDI agent, is publicly available at https://github.com/usnistgov/ARIAC. Any other data can be let available up on request.

Declarations

Competing interests The authors have no relevant financial or non-financial interests to disclose.

References

1. Harrison, W., Downs, A., Schlenoff, C.: The Agile Robotics for Industrial Automation Competition. AI Mag. **39**(4), 73–76 (2018)
2. Evjemo, L.D., Gjerstad, T., Grøtli, E.I., Sziebig, G.: Trends in smart manufacturing: Role of humans and industrial robots in smart factories. Curr. Robot. Rep. **1**, 35–41 (2020). https://doi.org/10.1007/s43154-020-00006-5
3. Marvel, J.A.: Performance Metrics of Speed and Separation Monitoring in Shared Workspaces. IEEE Trans. Autom. Sci. Eng. **10**(2), 405–414 (2013)
4. BBC-News: Man Crushed to Death by Robot in South Korea. https://www.bbc.co.uk/news/world-asia-67354709 (2023) Accessed 11 Nov 2023
5. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W.: Robot Operating System 2: Design, architecture, and uses in the wild. Sci. Robot. **7**(66) (2022). https://doi.org/10.1126/scirobotics.abm6074
6. Koenig, N., Howard, A.: Design and Use paradigms for Gazebo, an Open-source Multi-robot Simulator. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 2149–2154 (2004). https://doi.org/10.1109/IROS.2004.1389727
7. Wooldridge, M.: An Introduction to Multiagent Systems. John Wiley and Sons, New York, U.S. (2002)
8. Russell, S., Norvig, P.: Artificial Intelligence, A Modern Approach, 2nd edn. Prentice Hall, New Jersey, U.S. (2003)
9. Bond, A.H., Gasser, L.: Readings in Distributed Artificial Intelligence. Morgan Kaufmann, Massachusetts, U.S. (1988)
10. Bratman, M.E., Israel, D.J., Pollack, M.E.: Plans and resource-bounded practical reasoning. Comput. Intell. **4**(3), 349–355 (1988)
11. Davis, R., Smith, R.G.: Negotiation as a metaphor for distributed problem solving. Artif. Intell. **20**(1), 63–109 (1983)
12. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. Artif. Intell. **42**(2–3), 213–261 (1990)
13. Durfee, E.H., Lesser, V.R., Corkill, D.D.: Trends in cooperative distributed problem solving. IEEE Transactions on knowledge and data Engineering (1989)
14. Shoham, Y.: Agent-oriented programming. Artif. Intell. **60**(1), 51–92 (1993)
15. Rao, A., Georgeff, M.: Modeling rational agents within a bdi architecture, eds. r. fikes and e. sandewall. In: Proc. Second Conf. Knowledge Rep. Reasoning (Morgan Kaufman), pp. 473–484 (1991)
16. Bratman, M.: Intentions, Plans, and Practical Reason. Harvard University Press, Massachusetts, U.S. (1987)
17. Karaduman, B., Tezel, B.T., Challenger, M.: Rational software agents with the bdi reasoning model for cyber–physical systems. Eng. Appl. Artif. Intell. **123**, 106478 (2023). https://doi.org/10.1016/j.engappai.2023.106478
18. Dennis, L.A., Fisher, M.: Verifiable Autonomous Systems: Using Rational Agents to Provide Assurance About Decisions Made by Machines. Cambridge University Press, Cambridge, U.K. (2023). https://doi.org/10.1017/9781108755023
19. Fisher, M., Mascardi, V., Rozier, K.Y., Schlingloff, B., Winikoff, M., Yorke-Smith, N.: Towards a Framework for Certification of Reliable Autonomous Systems. Auto. Agents MultiAgent Syst. **35**(1), 8 (2021). https://doi.org/10.1007/s10458-020-09487-2
20. Bordini, R.H., Hübner, J.F., Wooldrige, M.: Programming Multi-Agent Systems in AgentSpeak Using Jason. John Wiley & Sons, New York, U.S. (2007)
21. Cardoso, R.C., Ferrando, A.: A Review of Agent-Based Programming for Multi-Agent Systems. Computers **10**(2), 16 (2021)
22. Buss Becker, L., Downs, A., Schlenoff, C., Albrecht, J., Kootbally, Z., Ferrando, A., Cardoso, R., Fisher, M.: Using a bdi agent to represent a human on the factory floor of the ariac 2023 industrial automation competition. In: Malvone, V., Murano, A. (eds.) Multi-Agent Systems (EUMAS 2023), pp. 214–227. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-43264-4_14
23. Correll, N., Bekris, K.E., Berenson, D., Brock, O., Causo, A., Hauser, K., Okada, K., Rodriguez, A., Romano, J.M., Wurman, P.R.: Analysis and Observations from the First Amazon Picking Challenge (2016)
24. Marvel, J.A., Hong, T.-H., Messina, E.: 2011 solutions in perception challenge performance metrics and results. In: Proc. of the Workshop on Performance Metrics for Intelligent Systems, pp. 59–63. ACM, New York, NY, USA (2012)
25. Rao, A.S.: AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. In: Agents Breaking Away, MAAMAW 1996. LNCS, vol. 1038, pp. 42–55. Springer, Berlin, Heidelberg (1996)
26. Tausch, A., Kluge, A.: The best task allocation process is to decide on one's own: effects of the allocation agent in human–robot interaction on perceived work characteristics and satisfaction. Cognit. Technol. Work **24**, 39–55 (2022). https://doi.org/10.1007/s10111-020-00656-7
27. Barenji, A.V., Barenji, R.V., Roudi, D., Hashemipour, M.: A dynamic multi-agent-based scheduling approach for smes. Int. J. Adv. Manuf. Technol. **89**, 3123–3137 (2017)

Leandro Buss Becker is currently on leave from the Federal University of Santa Catarina and is a senior postdoctoral researcher at the Computer Science Department of the University of Manchester. His research interests relate with the design of safety-critical embedded applications, with special interest in software engineering and BDI agents architectures within such a domain. Dr. Buss Becker joined different program committees from local and international scientific events in the area of his research interests. He is also a reviewer for several journals in these areas. He also chaired and co-chaired several scientific events, such as the Brazilian Symposium on Computing Systems Engineering (SBESC). He is co-founder of SBESC and, from 2014 to 2018, the was the head of the Brazilian Computer Society working group on Computing Systems Engineering (SBC/CE-ESC).

Anthony Downs (Tony) is the Project Leader for the Agility Performance of Robotic Systems project within the Measurement Science for Manufacturing Robots Program and is the Lead for the Agile Robotics for Industrial Automation Competition (ARIAC), an annual Prize competition currently run in simulation. He is the Chair for the IEEE Standards Association Working Group P2940 Measuring Robot Agility, where he is working with members from Industry, academia, and government to develop standards for measuring Agility focusing on Manufacturing environments and industrial robots. In the past, Tony has worked on the Urban Search & Rescue Robotics Standards project for DHS, as well as various DARPA Projects including DARPA ASSIST, DARPA TransTac Evaluations, and was one of the main application evaluators for the DARPA Transformative Apps (TransApps) project. He was a key player in running the Ultra-Wide Band (UWB) tracking system for both the Emergency Robots project as well as for Data Collections by the ARL CTA project. Tony is also the Division Safety Representative for the Intelligent Systems Division (735), and has recently completed 2 years on the EL Diversity, Inclusivity, and Belonging (DIB) Council as one of two division 735 representatives.

Craig Schlenoff is the Group Leader of the Cognition and Collaboration Systems Group and the Program Manager of the Robotic Systems for Smart Manufacturing Program in the Intelligent Systems Division at the National Institute of Standards and Technology. He leads the Agility Performance of Robotic Systems project and co-leads the Embodied AI and Data Generation for Manufacturing Robotics project. His research interests include knowledge representation/ontologies, intention recognition, and performance evaluation techniques applied to manufacturing robotic systems. He has led multiple million-dollar projects, dealing with performance evaluation of advanced military technologies and agility performance of manufacturing robotic systems. He has published over 150 journal and conference papers, guest edited three journals, and written three book chapters. He is currently the Associate Vice President for Standardization within the IEEE Robotics and Automation Society and the vice chair of the IEEE Robot Task Representation Working Group. He previously served as the program manager for the Process Engineering Program at NIST and the Director of Ontologies at VerticalNet. He received his Bachelor's degree from the University of Maryland and his Master's degree from Rensselaer Polytechnic Institute, both in mechanical engineering, and his PhD from the University of Burgundy, France in computer science.

Justin Albrecht is a Mechanical Engineer at NIST and member of the Agility Performance of Robotics Systems project. He recieved a Bachelor of Science in Mechanical Engineering and a Masters of Engineering in Robotics from the University of Maryland, College Park. Justin is a lead developer for the Agile Robotics for Industrial Automation Competition (ARIAC), an annual Prize competition. He led the transition of the simulation software from ROS1 (Robot Operating System) to ROS2. Justin also works on standards efforts with ASTM as a part of Commitee F45 for Robotics, Automation and Autonomous Systems, specifically on developing test methods for standardization of grasping and manipulation of robotic systems.

Zeid Kootbally received his Ph.D. in computer science in 2008 from the University of Burgundy, France, for a dissertation on "Moving Object Predictions in Dynamic Environments for Autonomous Ground Vehicles". Dr. Kootbally is currently a Computer Scientist at the National Institute of Standards and Technology (NIST), in the Intelligent Systems Division (ISD). Dr. Kootbally's research is focused on developing the measurement science for the evaluation of agile robotics systems applied to manufacturing. Dr. Kootbally is a contributor to the Agile Robotics for Industrial Automation Competition (ARIAC).

Angelo Ferrando is an Assistant Professor at the University of Modena and Reggio Emilia. He previously was a Research Fellow at the University of Genova (IT) and a Research Associate at the University of Liverpool (UK). Since 2021 to 2024, he has been the coordinator of the CINI "Smart Cities & Communities" node at the University of Genova. Previously, he was a Postdoctoral Research Associate at the University of Manchester (UK) from July 2020 to February 2021 and a Postdoctoral Research Associate at the University of Liverpool (UK) from 2018 to 2020. He was part of the Autonomy and Verification Network at the University of Manchester, where he worked on applying RV in the RAIN and ORCA research hubs funded by UKRI. He is a member of the program committee in several international conferences such as AAAI, AAMAS, and IJCAI. His main research interests are Formal Methods, Software Engineering, Multi-Agent Systems (MASs), and Runtime Verification (RV), especially when applied to improve the reliability of intelligent systems.

Rafael C. Cardoso is a Lecturer in the Department of Computing Science at the University of Aberdeen (UK), and a member of the Agents at Aberdeen (A3) research group and the Autonomy and Verification Network. He is also an Honorary Senior Research Associate at both the University of Liverpool (UK) and the University of Manchester (UK). Rafael obtained his Master and Doctorate degrees in Computer Science from PUCRS (2014 and 2018 resp.), and Bachelor's degree in Computer Science from Universidade Franciscana (2012). His main research themes include cross-cutting research on multi-agent systems, AI planning, verification and validation, and neuro-symbolic AI. Rafael is a member of the program committee in several top-ranked international conferences such as AAAI, IJCAI, AAMAS and ECAI, where he also publishes his work.

Michael Fisher is currently with and Computer Science Department of the University of Manchester and the Chair of emerging technologies with the Royal Academy of Engineering, London, U.K. His research interests mainly concerns autonomous systems, particularly verification, software engineering, and trustworthiness, with recent work including developing an automated verification system for agent programs, hybrid agent architectures for robotic systems, self-awareness, and formal verification for use in the certification of autonomous systems. He is a member of the British Standards Institution AMT/10 Committee on Robotics and the IEEE Standards Group for IEEE-P7009 on Fail-Safe Design of Autonomous and Semi-Autonomous Systems. He is on the Editorial Boards of both Applied Logic and Annals of Mathematics and Artificial Intelligence journals. He is also a Corner Editor of the Journal of Logic and Computation. He holds the European Chair position for the IEEE Technical Committee on the Verification of Autonomous Systems.

Authors and Affiliations

Leandro Buss Becker[1,2] · Anthony Downs[3] · Craig Schlenoff[3] · Justin Albrecht[3] · Zeid Kootbally[3] · Angelo Ferrando[4] · Rafael Cardoso[5] · Michael Fisher[1]

Anthony Downs
anthony.downs@nist.gov

Craig Schlenoff
craig.schlenoff@nist.gov

Justin Albrecht
justin.albrecht@nist.gov

Zeid Kootbally
zeid.kootbally@nist.gov

Angelo Ferrando
angelo.ferrando@unimore.it

Rafael Cardoso
rafael.cardoso@abdn.ac.uk

Michael Fisher
michael.fisher@manchester.ac.uk

1 Computer Science Dept, University of Manchester, Oxford Rd. – Kilburn building, Manchester M13 9PL, UK

2 Federal University of Santa Catarina, Florianopolis, Brazil

3 National Institute of Standards and Technology, 100 Bureau Dr, Gaithersburg, MD 20899, USA

4 University of Modena and Reggio Emilia, Modena, Italy

5 University of Aberdeen, Aberdeen, UK