DriveToGæther: a Turnkey Collaborative Robotic Event Platform

Florence Dupin de Saint-Cyr[1][a], Nicolas Yannick Pepin[2][b], Julien Vianey[1][c], Nassim Mokhtari[3], Philippe Morignot, Anne-Gwenn Bosser[3][d] and Liana Ermakova[3][e]

[1]*IRIT, Toulouse University, France* [2]*Zoetis, Danemark* [3]*Lab-STIIC, Brest, France*

Keywords: Learning by doing, cooperation AI/human, robots, ludic event

Abstract: This paper reports the organization of an event that enabled experts as well as non-specialists to practice Artificial Intelligence on robots, with the goal to enforce human-AI cooperation. The end aim of this paper is to make the material and virtual platform built for the event reusable by as many people as possible, so that the event can be reproduced and can give rise to new discoveries or to the production of new data sets and benchmarks. The underlying purpose is to de-demonize AI and to foster group work around a fun, rewarding and caring project.

1 INTRODUCTION

The aim of the Turing test was to test the credibility of a machine by checking whether it could be mistaken for a human, and thus to demonstrate the intelligence of the machine compared with that of the human. The goal of inventing a machine that can deceive a human being by imitating him or her no longer seems as interesting today, since on the one hand it has been successful in certain fields, and on the other because the future lies more in complementarity than in human/machine competitiveness. This complementarity is all the more necessary in fields where human performance is poor: there are around 3,600 deaths a year in traffic accidents on French roads, 40,000 in the USA and many more in developing countries. Although robotized vehicles were proposed in the early 90s as a way of improving this situation, society seems to be moving towards road traffic that is certainly made up of autonomous vehicles, but which will also coexist with cars driven by humans, at least initially.

This paper describes the organization of an event that enabled experts as well as non-specialists to practice Artificial Intelligence on robots with the ambition of making humans and AI work together. The workshop was open to anyone interested in developing efficient, cooperative, and adaptive algorithms: researchers, students, academics, high school students, engineers, hobbyists, industrialists, and the general public. Indeed the DriveToGæther event is a place where AI algorithms are neither in competition with humans nor tasked with deceiving them by pretending to be human. The material and virtual platform built for the event is intended to be reusable by everyone, so that the event can be reproduced and give rise to new discoveries or to the production of new data sets and benchmarks[1]. The end goal of this article is to make these resources visible and accessible.

Traditional robotics competition (like e.g. RoboCup or the French mapping challenge "CAROTTE") favors the integration of sensor and actuator algorithms into a robotic platform (with perception, SLAM[2], data fusion, control, etc.), it is the same for educative robotic platforms such as e.g. Duckietown (Paull et al., 2017). In contrast, the aim of the DriveToGæther event is to focus on robot intelligence, and in particular on the complementarity between artificial and human intelligence. More precisely, the goal is to integrate high-level algorithms within a robotic platform, highlighting the intelligence of robots for cooperating with human.

The event organizers supplied robots already programmed to obey simple commands to execute over a playground made up of carpet tiles (on which lines are painted): go ahead, turn left or right, pick up or drop off victims. Each robot, controlled either by a human or by a program has to integrate the configu-

[a] https://orcid.org/0000-0001-7891-9920
[b] https://orcid.org/0009-0001-3361-5381
[c] https://orcid.org/0000-0001-7897-5666
[d] https://orcid.org/0000-0002-0442-2660
[e] https://orcid.org/0000-0002-7598-7474

[1]The link to the github project containing all the source code and documentation is https://github.com/cecilia-afia/DriveToGaether.

[2]SLAM: Simultaneous Localization and Mapping

ration of the flat playground. Victims, hospitals, and starting positions for the robots are placed on the playground, marked by cards on the floor, and the configuration of the cards can change with each game (see Figure 1). Instructions must be sent to a central server, which acts as a filter to check their feasibility before they can be executed by the robots and announced to all the robots. The global goal is to save all the victims in the playground, i.e., pass over the victim cell, beep once (to signal pick-up), pass over a hospital cell, beep twice (to signal drop-off). Each robot is allowed to transport a maximum of 2 victims at a time.

Figure 1: DriveToGæther event in 2023

This paper is organized as follows: Section 2 describes existing approaches related to human-machine collaboration; then the genesis of DriveToGæther is presented toward the recall of the organization of several events from 2016 to 2022. The physical and virtual platforms used during DriveToGæther event, with all their components and the competition's rules, are depicted in Section 4; the organisational and scientific outcomes of the event are finally discussed.

2 RELATED WORK

In this section we are interested in reviewing some domains where AI should at least take into account and even cooperate with humans. Let us first start by evoking the idea of Centaur-AI by quoting (Case, 2018): "In 1998, Garry Kasparov held the world's first game of 'Centaur Chess' (...) half-human, half-AI. But if humans are worse than AIs at chess, wouldn't a Human + AI pair be worse than a solo AI? Wouldn't the computer just be slowed down by the human (...)? In 2005, an online chess tournament, inspired by Kasparov's Centaurs, tried to answer this question. They invited all kinds of contestants - supercomputers, human grand-masters, mixed teams of humans, and AIs - to compete for a grand prize. Not surprisingly, a Human + AI Centaur beats the solo human. But -amazingly- a Human + AI Centaur also beats the solo computer." This promising idea about Centaur AI is inspired from the nature where symbiosis has many positive effects: quoting again (Case, 2018): "Symbiosis shows us you can have fruitful collaborations even if you have different skills, or different goals, or are even different species. (...) Symbiosis is two individuals succeeding together not despite, but because of, their differences. Symbiosis is the "+"." Indeed as we shall see in the following related works, machine-AI teaming is very promising in many domains.

2.1 Intelligent transportation systems

In the domain of Intelligent Transportation Systems, a.k.a., the autonomous and connected vehicle, six levels of autonomy are defined, from vehicles with "no autonomy" (level 0) to "fully autonomous" vehicles (level 5) (Nashashibi, 2019), via level 1 where the driver is constantly in charge of the maneuvers but may delegate easy tasks, level 2 where the responsibility is entirely delegated to the system but constantly supervised by the driver, level 3 where the driver can do other tasks during the travel but must be able to take control back when conditions require it, level 4 where the driver can completely focus on other tasks.

These levels may vary during a trip with autonomous driving: the autonomous vehicle can adapt its autonomy to the state of the driver so that control has to be arbitrated between the driver and the system (Morignot et al., 2014). For this, the system perceives the driver's state by pointing a camera at him and detects the level of tiredness of the driver by analyzing his/her blinking frequency and head angle. If the driver is fully awake, the system can switch to level 0, for the driver to actually fully drive the vehicle. But if the driver is tired, levels 2 or 3 are chosen due to less focus capability of the driver. When a driver is exhausted and fully asleep, the system switches to automation level 5. Therefore, human driver and system collaborate towards avoiding collision while driving the vehicle.

2.2 Human-aware social robots

A major contribution to human-robot interaction is presented in (Lemaignan et al., 2017), which describes a cognitive architecture for social robots: robots that communicate with humans in a multimodal way (natural language, designate objects by gesture, etc) and collaborate towards achieving common tasks in domestic interaction scenarios, e.g., cleaning a table covered with objects, moving objects to a different home with a robot helping to pack.

This human-aware deliberative architecture is based on the principle that human-level interaction is easier to achieve if the robot itself relies internally on human-level semantics. Therefore this architecture is based on an active knowledge base: the geometric reasoning module produces symbolic assertions describing the state of the robot environment and its evolution over time. These logical statements are stored in the knowledge base, and queried back, when necessary, by the language processing module, the symbolic task planner, and the execution controller. The output of the language processing module and the activities managed by the robot controller are stored back as symbolic statements as well. This deliberative architecture has been implemented and tested in several scenarios (see (Lemaignan et al., 2017)).

2.3 Human-robot teaming

A major contribution towards human-robot interaction is represented by the man-machine teaming project, organized by Dassault and Thales companies, aiming at creating an aerial cognitive system (Dassault-Aviation, 2018). It consists in equipping the systems with greater autonomy towards collaborative work, that would make operator actions and decisions more efficient and optimized while mobilizing less of the operator's mental and physical resources. The topics cover diverse themes such as virtual assistant and intelligent cockpit, man-machine interaction, mission management system, intelligent sensors, sensors' services, robotized support, and maintenance.

Human-robot teaming requires the machine to recognize the goal of the human, this domain of research is very large, see e.g. (Van-Horenbeke and Peer, 2021) for a review of the field. Indeed as these authors say: "recognizing the actions, plans, and goals of a person is a key feature that future robotic systems will need in order to achieve a natural human-machine interaction". Furthermore the idea to not only help by guessing what the human will do but also to suggest the person to adopt a given behavior is also a subject of research called *cognitive planning* (see e.g. (Fernandez Davila et al., 2021) in a logic-based setting).

2.4 Interactive diagnosis

In the medical domain, (Henry et al., 2022) describes the results of qualitative analysis of coded interviews with clinicians who use a machine learning-based system for diagnosis. Rather than viewing the system as a surrogate for their clinical judgment, clinicians perceived themselves as partnering with the technology. Even without a deep understanding of machine learning, clinicians can build trust with an ML system through experience, expert endorsement and validation, and can use systems designed to support them.

Note that there were remaining perceived barriers to the use of ML in medicine: potential for overreliance on automated systems, risk to standardize automatic care even in scenarios where a clinician disagrees with the system.

3 The DriveToGæther genesis: from Human–toy-robots teams contests to a jam for text generation

This section recall the competitions organized from 2016 to 2022 in order to make cooperate toy-robots either with other toy-robots or with human-driven toy-robots and also the organization of a jam. The successive organization of these events leads us to propose the DriveToGæther event, that we will present in Section 4, which was nomore a competition but a jam for toy-robots and human cooperation.

3.1 2016 Ricochet robot contest

The first event organized by our group was a competition called "AI on robots" in 2016 where robots had to explore a flat playground of 10m $\times$ 5m, with a grid drawn on the floor and walls made of wood boards disposed on the borders of some cells of the grid (see Figure 2). The positions of the interior walls could vary and were unknown to the competitors before each game. Each team could bring and use several robots. The aim of each team was to send a robot to

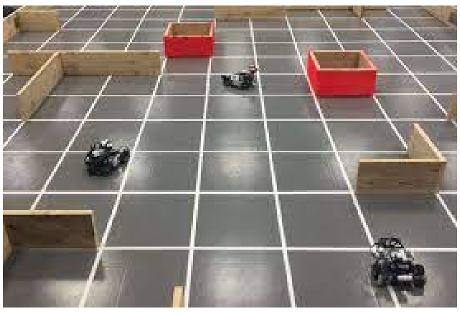

Figure 2: Ricochet robot contest playground

a destination. This was done in two steps: a phase for mapping the arena followed by a phase to get to the destination given the mapping. The imposed robots were Lego Mindstorm (NXT or EV3, see Figure 3).

Figure 3: A Lego Mindstorm NXT at Ricochet robot contest

For the second phase, the rules of the competition were adapted from the ”Ricochet Robot” game: from an entry point in the arena, reach a destination by ricochet moves: the robot always goes straight ahead and can only change direction when it “bounces off” an obstacle (a wall or another robot). Several levels of difficulty were proposed: depending on the number of teams in the arena (1 or 2) and depending on the information about the destination (given at the start or discovered when encountering another robot). In the latter case, the robot(s) in each team must first pass an opposing robot to find out their target destination, before trying to reach that destination by bouncing off obstacles.

In total the competition welcomed about fifty visitors, with the visit of a primary school class. The 12 participants of the 7 registered teams were researchers, PhD, Master and Bachelor computer science students. They were kind enough to answer the children’s many questions. A television crew came to film for the local news (France 3 Auvergne, 2016).

3.2 2019 DriveToGæther contest

In this contest, the participants should bring their robots, no specific type was imposed, the only constraint was that the robots do not exceed the size of the carpet tiles (see Figure 4).

A team had to control two robots, a robot directly piloted by a human and an automatic robot. The automatic robot had to integrate beforehand the configuration of the playground the places of the victims and hospitals and the starting positions of both robots.

Several challenges were proposed:

- build the most manoeuvrable robot (a questionnaire on the manoeuvrability was completed by the participants and the jury).
- be the fastest to save all victims on the playground
- be the most effective in number of operations on this same playground (the operations of each robot were counted, the maximum was returned).

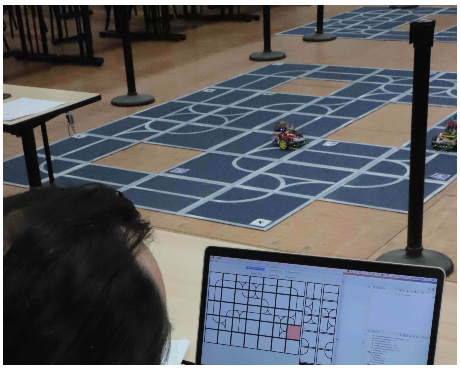

Figure 4: The 2019 DriveToGæther contest

AI used: One of the teams proposed to launch several A* to determine the shorter paths from the robot to the different victims (or hospitals). For this purpose, they used a “heat map” in which the cells close to other robots were considered hot (therefore weighted with a positive weight) and were repellent (heat depended on the proximity of robots), cold areas (weighted with a negative weight) were desirable (freshness depended on the proximity of hospitals or victims). The interface was coded in Java, the arduino was a kind of C++, and the embedded AI on Raspberry was encoded in Python.

A team made an interface on which a user can specify the playground (graph with vertices with positions of the victims, hospitals, starting points). From the interface, it generates a PDDL representation of the problem (composed of an initial state and a goal) that is solved with CBT which is an optimal anytime planner (whose quality increases over time). For example by leaving 2 minutes to the algorithm, it obtained a solution for a single robot in 32 operations (for the small playground with 6 victims and 2 hospitals). From the obtained plan, the first action was executed then the planner was relaunched from the new state as long as there were still victims to save. The development was done in Java.

Another team assumed that the human would choose the best rescue strategy. They also chose to rely on humans to avoid immediate collisions. Thus, an algorithm estimates the most likely human decision and translates it in terms of proximity to accessible victims/hospitals. The algorithm calculates a possible complete strategy of the human. It then chooses among the human strategies of equivalent probability the farthest goal to reach for the human-driven robot which is closest to the automatic robot. Therefore the latter realizes the strategy complementary to that of the human beginning with the last actions of it. After each action there is an estimate of the deviation from the estimated human strategy and re-calculation of the possible strategies of the human if it has deviated. Development was done in Python.

Outcomes of the competition In total the competition welcomed about forty visitors. Given the amount of work to achieve the material and regulatory framework of the competition, given the interest of the challenge itself expressed by the public and participants, the competition was worth replicating. However we thought that we should find a way to encourage more participation: we faced a lot of last-minute withdrawals from local students not available on the dates of the contest although having realized the necessary programs (during internships).

The very good results obtained by the team that was not prepared before, and that registered on site only, made us consider that a **Game jam** format or **Hackaton** could bring more participation.

3.3 2022 Text Generation Jam

The *Jam Generation of texts which are poetic or fun or both*, event was organized in 2022 (Bosser et al., 2022). Inspired by Game Jams[3] and Proc Jams[4], this event gave participants the chance to play together to implement AI tools with the same goal in mind. A jam is a ludic event where creativity is put forward. It is a time to meet and a place to experiment, where sharing skills and learning new technologies are encouraged.

The participants were provided with research papers on humor and poetic generation (He et al., 2019; Weller et al., 2020; Van de Cruys, 2020; Valitutti et al., 2013), a list of AI models as well as data, such as Lexicon 3 [5], ConceptNet [6], the JOKER corpus of wordplay in English and French (Ermakova et al., 2023). Specific themes or constraints on the generated text were proposed by the organizers which were addressed by a range of AI techniques applied by the participants (Bosser et al., 2022).

Forty participants were attracted by the jam. The enjoyment of an artificially created joke is subjective and may vary from person to person. Evaluating the creativity of Large Language Models LLMs is a difficult task, and when addressing the challenges associated with jokes produced with minimal human involvement, we should take into account the human lack of trust towards the actual creativity of the AI.

[3]https://globalgamejam.org/
[4]https://www.procjam.com/
[5]http://www.lexique.org/
[6]https://conceptnet.io/

4 PLATFORM DESCRIPTION

The event organized in 2023 by our group dealt with AI on robots following the competition named DriveToGæther organized in 2019. Hence, the challenge is for robots to ”rescue” all victims present in the playground by ”picking them up” and ”dropping them off” at hospitals (these two actions are performed virtually, with only a sound emitted to signify that an action has been performed). This rescue can be carried out by robots controlled by humans and by robots controlled by autonomous programs, bearing in mind that autonomous robots only have access to:

- the initial positions of the other robots;
- victims and hospitals positions;
- playground configuration;
- movements carried out by other robots.

Due to the difficulty, encountered during the 2019 DriveToGæther contest, of motivating people to participate and prepare their own robots in advance, we decided to organize the event in the same spirit as the *Jam Generation of texts which are poetic or fun or both*, (Bosser et al., 2022), described above. Moreover, in order to make the event easier to organize, we proposed to develop a central server to which all information is transmitted, this server is in charge of managing automatic refereeing, the robots' actions on the playground, as well as the entire game progression. The game is described by the rules referred below, it involves several tasks.

- the development of a server enabling clients to play and/or follow games,
- development of a client (agent observer) displaying a virtual dashboard,
- development of a basic client (agent driver) that is able to communicate with the server
- set up a fleet of physical robots (mbots) that obey the server.

4.1 Rules

The aim is not only to produce autonomous robots, but also to get these computer-controlled robots to work together with human-driven robots and towards a common goal. The robots symbolize cars on the move in a city. In this city, there are both autonomous cars (controlled by a computer program) and cars driven by humans. Unfortunately, city dwellers can suffer illness or accidents, in which case they are called victims, and it is necessary to get all the victims to hospitals and do this as quickly as possible while avoiding collisions with other rescuers. An example of city map made up of lines to follow is described on figure 6. There are three types of tiles (all of square size (50x50 cm)) as shown in Figure 5.

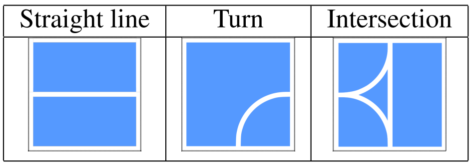

Figure 5: The three types of tiles

There are three types of zones of interest: robots start positions, victim positions and hospital positions. These positions are only given at the start of each match. The rules of the games contain the following constraints:

1. A robot is considered to occupy a cell when part of it is on the carpet tile corresponding to this cell.
2. A robot must not enter an occupied cell.
3. The robots follow the lines and never stop between two cells: any instruction makes it either to stay in its cell or to entirely move into another distinct cell.
4. The robots signal when they have picked up a victim and when they have dropped their victims off with one/two beep(s) respectively. The robots have a limited carrying capacity of at most two victims at a time.
5. The (human or autonomous) driver does not have direct access to the robots. The driver should send instructions to a central server that will move the robot, only if authorized. The instructions are:
    - Move (go straight ahead): requires the robot to be on a cell that is not an intersection
    - Right (take the most right line): requires the robot to be at an intersection
    - Left (take the most left line): requires the robot to be at an intersection
    - Drop (drop off all the victims): requires the robot be transporting victims
    - Pick (pick up a victim): requires a victim on the cell where the robot is
    - NOP (do nothing)

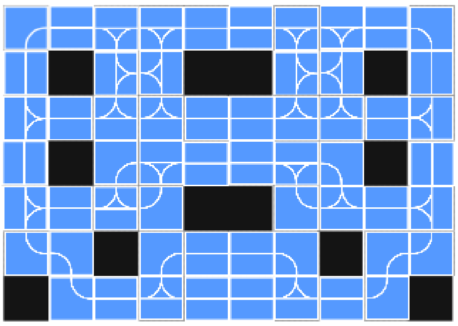

Figure 6: An example of playground map

4.2 The physical platform

Our equipment consisted of 60 carpet tiles that we painted with turns, straight lines and intersections, 6 *mbots* with 6 *Raspberry Pi 3*, plugged with their own batteries and sticked on the mbots roofs. The commands sent by the central server were transmitted to the Raspberry Pi via an internet connexion and then transmitted to the mbot via a wired connexion (see Figure 7).

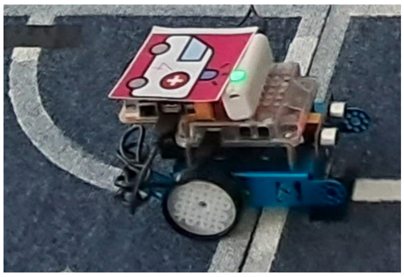

Figure 7: Mbot with Raspberry Pi 3 and battery on its top

The mbots and the Raspberry Pi are programmed in Arduino Language, mbots are programmed to wait for instructions and to move accordingly, while following the lines, and to stop in the next carpet tile. The Raspberry Pi are programmed to transmit instructions to the mbots. Both implementations use the "SoftwareSerial" library (which simulates the functionalities of a micro controller card).

The central server is responsible for:

- loading the initial configuration of the playground (an interface allows the user to enter the number of players, the number of robots, the start positions of the robots, the places of victims and hospitals, the dimensions of the playground and the configuration of the tiles),
- checking if the command sent by a (human or automatic) driver towards a robot is feasible,
- transmitting validated orders to robots and inform all users of the actions successfully completed,
- showing publicly the current state of the game.

4.3 The virtual platform

The virtual platform developped in Python during the event is shown in Figure 8, it allows the user to simulate the movement of agents in a configurable environment. Two essential configuration files, `terrain.txt` and `config.txt`, are used to define the placement of agents, victims, and hospitals:

- `terrain.txt`: Specifies the terrain's rectangular shape and employs specific characters to define elements such as obstacles.
- `config.txt`: Describes the initial position of each agent, using the same dimensions as `terrain.txt`. Each agent is identified by a number and an orientation.

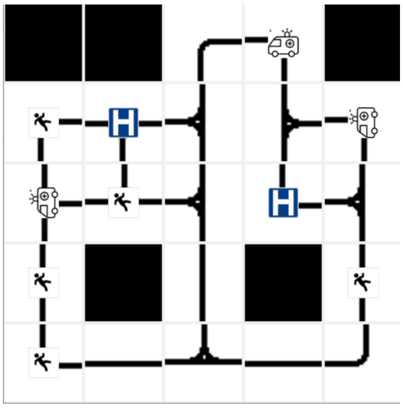

Figure 8: Virtual platform

The simulation is managed by the *Robot* and *Simulator* classes, with a graphical user interface (GUI) for real-time visualization. The *Robot* class represents the agents in the simulated world. `Robots` have attributes such as their position and orientation. The `doAction()` method is essential for a robot to randomly choose from the six possible actions (Drop, Pick, Move, Left, Right, Uturn). When feasible, this method prioritizes Drop and Pick over the other actions. In the absence of any available action, the `doAction()` method will opt for Nop (No Operation). The *Simulator* class manages the evolution of the simulated world and provides a graphical interface for real-time simulation visualization. It handles robot movement, victim picking and dropping, and verifies the validity of actions taken by robots through the `check_action()` method. If an action chosen by a robot is incorrect, it returns Nop.

This simulator can potentially be used for communication with physical robots and for managing their movements through its `check_action()` method. Furthermore, its graphical user interface (GUI) can serve as a visual tool for monitoring the progression of the robots within the environment.

5 RESULTS AND DISCUSSION

Around fourty persons participated to this jam, the playground was open from 10 a.m. to 6 p.m. during 5 days at the heart of a national conference (attended by nearly 500 persons). Among the results produced by the jam participants we have:

- a visual interface for editing a playground,
- a simulator for moving robots on the playground,
- a driver based on simplex algorithm,
- a driver based on goal ordering with Dijkstra,
- a driver programmed with reinforcement learning.

To sum up, we have successfully elaborated a platform enabling to test whether a robot (and more precisely the program driving this robot) is intelligent. By "intelligent", we do not mean here that robots would pass a Turing test, but that robots are capable of intelligently collaborate with humans, without receiving orders nor anything else than indications about the performed motions of the other robots. Here are the advantages of our platform:

- open source for educational or research purposes: our platform acts as a simulator for cooperative planning, reinforcement learning and other A.I. approaches,
- reproducible thanks to the description given here and on github,
- accessible: mbots, raspberry Pis, carpet tiles are cheap and readily available on the market,
- our event is not a competition, it is a jam were everyone can contribute,
- our platform considers A.I. as a tool to help humans: A.I. is not a rival nor a lier.
- our platform is applied and concrete, with important and easy to understand challenges,
- it is easy to use both for the organizers and participants (who write source code on site) hence have a light preparation work,
- the platform success can be quantified in terms of number of participants, but also in terms of variety of algorithm ideas.

We claim originality since most organized events about robotics are competitions that focus on the candidate planners ability to solve each problem and on their speed — not on their adaptability to humans. Take for instance the ICAPS conference series[7], it is a

[7] International Conference on Automated Planning and Scheduling, see the last event at https://icaps23.icaps-conference.org/

forum dedicated to planning and scheduling research, and it includes competitions among planners[8] that are made more for enhancing performances of robots than for increasing human-robot cooperation.

Note that, we do not exclude to take part to some competitions, thanks to the development done by our jam members, indeed a future application of this work could involve a participation to e.g. the Urban Challenge and the Grand DARPA Challenge, in which real robotized vehicles must find their way in a city or in a desert to reach a final point: with real vehicles, real obstacles and real goals to achieve.

More generally, we aim at evolving towards a collaborative game in which robots need to collaborate to reach goals (e.g. to enable an access to knowledge, or to a treasure, or to take pictures of monsters) while guessing other robots intentions (see (Gesnouin, 2022)). This evolution would be included in the challenges that we want to organize under the form of open jams, with in mind the idea of getting everyone involved to help drive AI forward. Ultimately, the biggest challenge is to get humans and machines to work together, taking advantage of the machine computational capabilities (good at answering questions) and human imagination (good at asking them), to create a fruitful cooperation, a centaur-AI.

REFERENCES

Bosser, A.-G., Ermakova, L., Dupin de Saint Cyr, F., De Loor, P., Charpenay, V., Pépin-Hermann, N., Alcaraz, B., Autran, J.-V., Devillers, A., Grosset, J., et al. (2022). Poetic or humorous text generation: jam event at PFIA2022. In *13th Conference and Labs of the Evaluation Forum (CLEF 2022)*, pages 1719–1726. CEUR-WS. org.

Case, N. (2018). How to become a centaur. *Journal of Design and Science*.

Dassault-Aviation (2018). Human machine interface. Technical report. [9].

Ermakova, L., Bosser, A.-G., Jatowt, A., and Miller, T. (2023). The JOKER Corpus: English–French parallel data for multilingual wordplay recognition. In *Proc. 46th Int. ACM SIGIR Conf. on Research and Dev. in Information Retrieval*.

Fernandez Davila, J. L., Longin, D., Lorini, E., and Maris, F. (2021). A simple framework for cognitive planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):6331–6339.

France 3 Auvergne (2016). Une compétition de robots en Lego dotés d'IA à clermont-ferrand. https://france3-regions.francetvinfo.fr/auvergne-rhone-alpes/puy-de-dome/clermont-ferrand/competition-robots-lego-dotes-intelligence-artificielle-clermont-ferrand-1035983.html.

Gesnouin, J. (2022). Analysis of pedestrian movements and gestures using an on-board camera to predict their intentions. Technical report, Mines.

He, H., Peng, N., and Liang, P. (2019). Pun Generation with Surprise. In *Proc. Conf. N. American Chap. of Assoc. for Comput. Linguistics: Human Language Tech.*, pages 1734–1744.

Henry, K. E., Kornfield, R., Sridharan, A., Linton, R. C., Groh, C., Wang, T., Wu, A., Mutlu, B., and Saria, S. (2022). Human–machine teaming is key to ai adoption: clinicians' experiences with a deployed machine learning system. *Nature, Digital Medecine*, 5:97.

Lemaignan, S., Warnier, M., Sisbot, E. A., Clodic, A., and Alami, R. (2017). Artificial cognition for social human-robot interaction: An implementation. *Artificial Intelligence*, 247:45–69.

Morignot, P., Rastelli, J. P., and Nashashibi, F. (2014). Arbitration for balancing control between the driver and adas systems in an automated vehicle: Survey and approach. In *2014 IEEE Intelligent Vehicles Symp. Proc.*, pages 575–580. IEEE.

Nashashibi, F. (2019). White paper on the autonomous and connected vehicles (in French). Technical Report [10], INRIA.

Paull, L., Tani, J., Ahn, H., Alonso-Mora, J., Carlone, L., Cap, M., Chen, Y. F., Choi, C., Dusek, J., Fang, Y., et al. (2017). Duckietown: an open, inexpensive and flexible platform for autonomy education and research. In *2017 IEEE Int. Conf. on Robot. and Autom. (ICRA)*, pages 1497–1504.

Valitutti, A., Toivonen, H., Doucet, A., and Toivanen, J. M. (2013). "Let Everything Turn Well in Your Wife": Generation of Adult Humor Using Lexical Constraints. In *Proc. 51st Annual Meeting of Assoc. for Comput. Linguistics*, pages 243–248.

Van de Cruys, T. (2020). Automatic poetry generation from prosaic text. In *Proc. 58th Annual Meeting of Assoc. for Comp. Linguistics*, pages 2471–2480.

Van-Horenbeke, F. A. and Peer, A. (2021). Activity, plan, and goal recognition: A review. *Frontiers in Robotics and AI*, 8:643010.

Weller, O., Fulda, N., and Seppi, K. (2020). Can humor prediction datasets be used for humor generation? Humorous headline generation via style transfer. In *Proc. 2nd W. on Figurative Language Processing*, pages 186–191.

[8] https://icaps23.icaps-conference.org/competitions/

[9] https://www.dassault-aviation.com/en/group/about-us/innovation/artificial-intelligence/human-machine-interface/

[10] https://www.inria.fr/en/white-paper-inria-autonomous-connected-vehicles