Sous Chef? Quickly Teaching Food Preparation Tasks to an Autonomous Robot Team

Matthias Scheutz[1,2], Eric Wyss[2] and John Peterson[2]

Abstract— **Autonomous robots have reached the physical capabilities to perform useful tasks in mixed-initiative human-robot teams. In this paper we provide an overview of the latest version of our DIARC architecture and show how it can be used in practical setting to quickly instruct a group of robots to perform novel tasks, in this instance food preparation tasks. Specifically, we motivate essential architectural capabilities required for effective mixed-initiative human-machine teaming and briefly discuss how DIARC meets them. We then introduce as an example of practical application a multi-robot, multi-human food assembly task as it occurs in fast food restaurants and show how different aspects of the task can be quickly taught through natural language dialogues, with the system immediately able to perform those tasks.**

I. Introduction

As robots are becoming more advanced in their manipulation and interaction capabilities, new applications are within reach that require robots to perform manual tasks as part of a mixed-initiative human-robot team (e.g., [1]). Here we focus on the fast food restaurant domain where humans are supported by robots in the preparation of food (e.g., [2]). The aim is to present and discuss an advanced robot application that utilizes a cognitive robotic architecture for seamless human-robot collaboration in a setting where robots can take over mundane, repetitive, and potentially inhumane tasks. While over the last couple of years several types of food preparation robots have been showcased at various technology venues (including bartender robots at IROS and CES with limited interaction capabilities), the robot system we present here enables natural spoken task-based dialogue interactions between multiple humans and multiple robots. The robots themselves employ advanced planning capabilities to handle task changes on the fly rather than executing pre-programmed or pre-learned tasks like those demonstration systems do.

The paper is organized as follows: We start by introducing the notion of "multi-human, multi-robot task-based interactions" that include not only natural language dialogues, but also joint actions in addition to anticipatory and supporting actions on both the human and the robot side. They also include developing *shared mental models* (e.g., [3]) that allow both humans and robots to track the overall task state, as well as encompass multi-goal dynamic settings where different team members work in parallel towards the accomplishment of multiple goals that require different resources and team members for completion. We then introduce the employed cognitive robotic architecture, followed by a detailed introduction of the application setting, the task and collaboration requirements, and how the architecture handles them. Finally, we demonstrate the various advanced features with traces from several interactions in a "mock" food preparation.

[1]Tufts University, Human-Robot Interaction Laboratory, Medford, MA, USA matthias.scheutz@tufts.edu

[2]Thinking Robots Inc., Boston, MA, USA eric@thinkingrobots.ai and pete@thinkingrobots.ai

II. Motivation

Many current and future team task settings involve the interaction and collaboration of multiple humans. In particular, they require the following two core capabilities: (1) **task-based natural language dialogue** — team members need to be able to have dialogues about task-relevant aspects, including goal priorities, execution status, performance estimates, task modifications and new goals, as well as unexpected task interference and ways to address them; (2) **anticipatory and supporting actions** — team members must be able to track task execution progress and determine when they need to perform actions in the interest of advancing team goals, including helping other team members with their subtasks, or enlisting the help of others explicitly when they cannot perform task steps on their own.

To enable the above for robots in mixed-initiative human-robot teams, the robot control architecture needs to meet several functional requirements. First, natural language dialogue capabilities need to be integrated into the architecture in a way that allows for introspective access to the robot's perceptions and actions, as well as its internal planning and reasoning components (otherwise the robot cannot engage in task-based dialogues). Second, in order to take appropriate actions, either reactive or anticipatory, the architecture needs to be able to develop plans for task execution and track their execution states. The plans might require the tracking of multiple parallel execution traces that involve any number of other robots and humans. Third, the architecture must allow for building and maintaining a *shared mental model* of the task and team states that is in sync with the mental models of the various human team members (otherwise robots will not be able to meaningfully engage in task-based dialogues or take task-relevant actions). Fourth, the architecture must also have the ability to adapt to dynamic changes in the task environment such as lack of availability of resources, changes in task assignments, as well as changes in goals and other task-related aspects. Finally, fifth, the architecture must be able to cope with unexpected changes to task environments that are likely to occur in "open-world settings" and must be able to accommodate various types of novelties on the

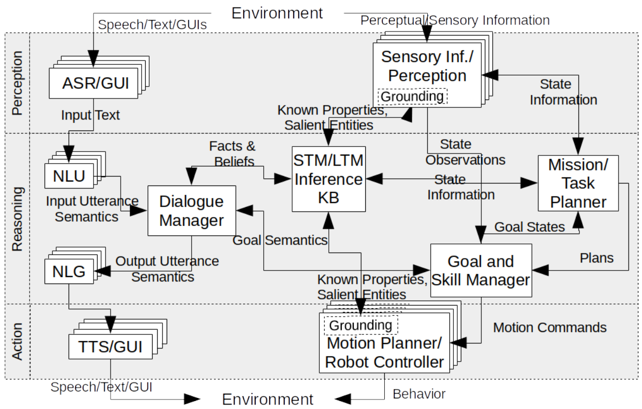

Fig. 1. The cognitive robotic DIARC architecture used for the development of restaurant application (see text for details).

fly, such as novel objects with new linguistic labels, novel procedures that require novel skills, new team members, new interaction processes, new events, and so forth [4]. While different environments will require different levels of sophistication depending on their level of "openness" (e.g., a warehouse environment will have fewer unexpected changes compared to an urban city street environment), the robot or robot team will at least need to have the ability to recognize unknown entities or task execution impediments, even if it is not able to cope with them.

III. The DIARC architecture

Here we will provide a brief overview of the *Distributed Interactive Autonomous Reflection Cognition* (DIARC) architecture (e.g., [5], [6]) and show how its way of implementing a "shared mental model" enables effective "multi-human, multi-robot task-based interactions".[1]

Fig. 1 shows one version of the overall architecture layout where white boxes depict architectural components that are implemented as middleware agents that can operate in parallel to other such agents and move freely within the employed TRADE middleware system (TRADE is an extension of the ADE system [8] with improved multi-robot interaction and introspection features). Component names indicate their functional roles in the architecture. Arrows depict information flow among architectural components with text on arrows describing the type of data that is sent along. Stacked components indicate that multiple instances of the same component type can exist within the same architecture. This is particularly useful for teaming contexts where multiple robots need to be operated and where multiple human team members need to be able to interact with the robots. Specifically, each robot will have its own "Sensory Information/Perception" component and its own "Motion Planner/Robot Controller" component, both of which typically run on the robot platform as they need to have access to the robot's hardware devices (sensors and motor controllers). Similarly, on the human interactant side, there will be one one "ASR/GUI" (Automated Speech Recognizer/Graphical User Interface") instance tied to the particular human to allow the system to identify the speaker and there will be one "NLU" (Natural Language Understanding) component responsible for generating natural language semantics. There will be one "NLG" (Natural Language Generation) component to convert internal semantics into human-understandable utterances that can be synthesized with a "TTS" (Text-To-Speech) component or displayed on a "GUI". The remaining component types — the "Dialogue Manager", the "STM/LTM Inference KB" (Short-Term Memory, Long-Term Memory, Inference Knowledge Base), "Mission and Task Planner", and "Goal and Skill Manager" — have only *one instance* in the system to be able to implement a "shared mental model" [3] that allows the robot cohort to function as a spatially distributed single entity that keeps track of the current task and teaming states. This particular way of implementing the shared mental model allows the architecture to share important task-based aspects among the robot cohort:

- **Dialogues:** Each robot will be able to have access to all dialogues with any other robot allowing the system to understand what kinds of requests different human teammates might have and how to most effectively respond to them. For example, a robot interacting with a human might not be able to obtain the answer to a posed question like "Do you know if the door to the kitchen is shut?" but another robot might, in this case a robot that can perceive the kitchen door. It is then possible for the local robot to consult the remote robot automatically and obtain the answer, which is then committed to the share knowledge base.
- **Shared knowledge base:** Sharing knowledge is particularly useful for allowing robots to work effectively in teams. For example, if one robot learns a new task, all robots will immediately know how to do it (e.g., [9]). Similarly, by updating the shared knowledge base with task-relevant information, the robot cohort will always have a comprehensive understanding of the current task state that it can act on (e.g., planning joint actions or redistributing subtasks).
- **Goal manager:** The common goal manager maintains the task breakdown into subgoals and all goal assignments to different agents, and can manage the robots' activities based on task and performance changes, as well as incoming instructions. For example, a robot might be tasked to perform a high-priority goal, in which case the execution of its current task will be paused. If there is a need to finish it, the goal manager can find another robot to continue the first robot's task.
- **Task planning:** The common task planner is particularly useful for determining interdependent plans based on the common knowledge of the robot cohort. For example, if it turns out that a new food item is needed

[1]Note that we slightly modified the previous acronym DIARC, replacing "Integrated" with "Interactive" and "Affect" with "Autonomous", to better reflect recent instantiations of the architecture compared to early versions such as [7].

and the goal manager has determined which robot should fetch it, the common planner can find a plan for the robot to get the item and possibly generate modified plans to the other robots impacted by the change. The task planner can also generate anticipatory actions such as fetching an item and bringing it to a human interactor if it is known based on the task representations in the knowledge base and goal manager's tracking of the human task performance that the human will need the item later on.

The various components of the DIARC architecture enables the above five architectural requirements for "multi-human multi-robot task-based interactions" by way of how the various functional components interact. To demonstrate its capabilities and readiness for applications, we will use a robot "sous chef" system as a running example to demonstrate.

IV. A Robot "Sous Chef"

We picked a food preparation task as it typically occurs in the kitchen of fast food restaurants to demonstrate the readiness of DIARC for practical applications (although the features discussed and demonstrated here are applicable to a much wider range of tasks and settings). In this task, multiple robots collaborate with multiple humans in preparing food made to order. Specifically, we consider a teaming setting with two robots — one stationary, one mobile — as well as several humans with different roles. One person serves as the *task instructor* who can teach the robots the relevant task information through natural language dialogues. Once the task has been instructed, other human team members can interact with the robot system to submit orders, provide missing ingredients, and deliver prepared foods to customers.

To allow for successful task instructions, the instructor needs to know the basic capabilities of the robots, including their perceptual capabilities (e.g., whether they know about and can recognize different types of foods, tools, etc.), their existing task-relevant knowledge (such as the purpose of cooking utensils, where they are located, etc.), and their skills (e.g., how to flip a burger with a spatula).

A. Knowledge representation and action primitives

At the very bottom level of the representational hierarchy, the robot uses *poses* (i.e., positions and orientations in end-effector space), *locations* (i.e., positions and orientations of objects in a world-centric map), and *objects* (i.e., simple representations of clustered regularities in the sensory data that can be mapped in end effector and world-centric coordinates). Poses can be used for a variety of actions; however, one of their central uses is to provide the positions from which visual searches can be run to find target objects that the robot needs to manipulate (as we assume the robots have cameras mounted on their end effectors). Examples of pre-defined action primitives are "savePose(name)" or "gotToPose(name)" for poses and "lookForObject(object)", "pickUpObject(object)", and "putDownObject(object,pose)" for object-based actions. Similarly, navigation behaviors are based on the same interaction idea whereby a human can teach named positions in the environment by making the robot follow it to a location using "followMe()" and storing its name using "saveLocation(name)". Subsequently, the robot can be instructed to go to a learned location using "goToLocation(name)". Additional skills, specific to the food preparation task, might include generic cooking skills "cook(object, numFlips, duration)" or frying skills "fry(object, duration)". Finally, at a higher level of the representational hierarchy, the robot needs domain-specific knowledge about the ingredients, which are used in recipes to prepare food items which, in turn, can be assembled to put together meals that can be delivered to the customer.

V. Teaching Food Preparation Tasks from Scratch

We consider a robot system consisting of an ABB Yumi mounted in a fixed location in the food preparation area, and a mobile platform for delivery. Each robot has a motion planner that can move its end effector to a desired position and orientation. The perceptual capabilities of the robots in this demo application are based on the functionality of Cognex SoC vision sensors which allow robots to capture an image, run a classification process on in, and translate the results to coordinates in its frame of reference (the Cognex cameras are mounted on the end effectors so they can be moved to different locations to run visual searches). Most interactions with the robots occur in natural language and can be dynamically expanded through the teaching of new poses, locations, objects, ingredients, items and meals.

The instruction of the food preparation task consists of three steps: (1) teaching locations in the environment, (2) teaching task-relevant objects like ingredients and where they are located, (3) teaching recipes for preparing food items.

A. Interactively teaching locations

The instructor walks through the space asking the robot to follow along and learn the names of different locations.

```
H: Robot one, follow me.
R: Okay, following.
H: Stop following.
R: Okay.
H: Save this location as 'delivery'.
R: Okay.
```

In addition to locations the robot can navigate to, the instructor can teach the robot poses and link them to locations, for example, teaching the "delivery table" pose which is associated with the delivery location.

```
H: Save this pose as 'delivery table'.
R: Okay. What location is this pose reachable
   from?
H: Location delivery
```

B. Teaching task-relevant objects

Once locations are defined, the instructor can teach new ingredients like "salsa verde", describe where they are located, what type of ingredient they are, and the name of the visual routine to detect them.[2]

```
H: Define new ingredient 'salsa verde'.
R: Okay. Where is it located?
H: Pose pantry.
R: What type of ingredient is it?
H: Type salsa.
R: What job is used to detect it?
H: Job detect salsa verde.
```

C. Interactively teaching food preparation and meal assembly

Once ingredients are defined, the preparation process itself can be instructed using these defined ingredients, poses, and locations, e.g., how to make fries.

```
H: Define new item 'fries'.
H: First, get a tray to prep area one.
H: Then get the fries box on the tray.
H: Then get the potatoes to the frier.
H: Then fry the potatoes for 90 seconds.
H: Then get the potatoes in the fries box.
H: That is how you prepare item french fries.
R: Okay.
```

Other items like "hamburger" can be instructed using an interaction template that allows the robot to ask the instructor specific questions.

```
H: Define new item hamburger
R: Okay. How many options does a hamburger
   have?
H: No options.
H: First get a tray to prep area one.
H: Then get a bun on the tray.
H: Then split the bun.
H: Get a lettuce on the bottom bun.
H: Get a tomato on the lettuce.
H: Then get a beef patty to the cook top.
H: Then cook the beef patty for 2 minutes
   flipping every thirty seconds.
H: Get the beef patty on the tomato.
H: Get mustard on the beef patty.
H: Get ketchup on the beef patty.
H: Get the top bun on the beef patty.
H: That is how you prepare item hamburger.
R: Okay.
```

VI. The "Sous Chef" System in Operation

Equipped with the initial knowledge, the robot system can immediately start operation, e.g., preparing a hamburger. At any time, any ongoing activity can be interrupted, e.g., by saying "pause current goal", and the system can be given more urgent tasks (e.g., to prepare a serving of fries), after which the previous task can be resumed. Most importantly, the robots' existing high-level task knowledge can be used to more quickly create new knowledge and procedures. Here, we demonstrate the whole meal preparation task with an ABB Yumi robot which has two arms, each equipped with a Cognex camera, that are completely independently controlled as would be the case with, say, two separate ABB GoFa arms, which shows how the system can work with multiple independent robots "as a system" but also as "individual parts", i.e., the architecture treats the two arms on the Yumi as a "team of two robots": "left arm" and "right arm". These team members can be instructed directly, or the robot can be addressed as a whole. [3]

We now show how the system can be instructed using task-based templates (e.g., [5]) that allow for instructing different versions of a meal (e.g., "southwest bowl with chipotle sauce") that the task planner can handle:[4]

```
H: Define new item southwest bowl.
R: Okay. How many options does a southwest
   bowl have?
H: One option.
R: What is option one?
H: The type of sauce.
R: How do I prepare a southwest bowl?
H: First get a serving box to serving area.
   Then get a bell pepper to the cook top.
   Right arm, saute the bell pepper for
   20 seconds.
   Then get the bell pepper in the serving box.
   Then get a corn to the hot plate.
   Then right arm cook a corn for 15 seconds.
   Then get the corn in the serving box.
   Then get a carrot to the cook top.
   Right arm, saute the carrot for 30 seconds.
   Then get the carrot in the serving box.
   Left arm, drizzle on the sauce.
   That is how you prepare a southwest bowl.
R: Okay.
```

Note that the above interaction shows how the instructor can give specific instructions that need to be carried out by a specific robot, in this case either the right or the left arm, as opposed to only instructing the "system" and letting the system determine which actions to perform with what arm depending on the task and how well-positioned a task is (Fig. 2 shows some snapshots from the task performance).

When the robot subsequently gets the instruction to prepare a "southwest bowl with chipotle sauce" (as an option), it can immediately start the task, but whenever it needs ingredients that are not available in its workspace, it will ask for help, e.g., "Could you please bring a bell pepper from the pantry to the prep area?". When the meal is complete, the robot announces it: "R: Southwest bowl is complete."

For the operation in the restaurant, the system needs to be able to distinguish between different types of users such as employees that are allowed to teach the robot information and customers who are allowed to order meals. For example, when an unauthorized user attempts to define a new meal like a "Puerto Rican bowl", the robot rejects the attempts, while an authorized user can define the new meal, in this case using *teaching by analogy*, a method that saves time by allowing

[2]This part is specific to the Cognex. With other sensors the object appearance could be also taught in one shot.

[3]Note that we used only one robot for the demonstration here to show the interactions with the human instructor who is taking on the task of supplying ingredients to the robot, but the system is perfectly general and another mobile robot could easily fill the role of providing the ingredients, e.g., see the demo in [5].

[4]A video showing the interaction can be found at `https://youtu.be/HzGzae8ccHA`.

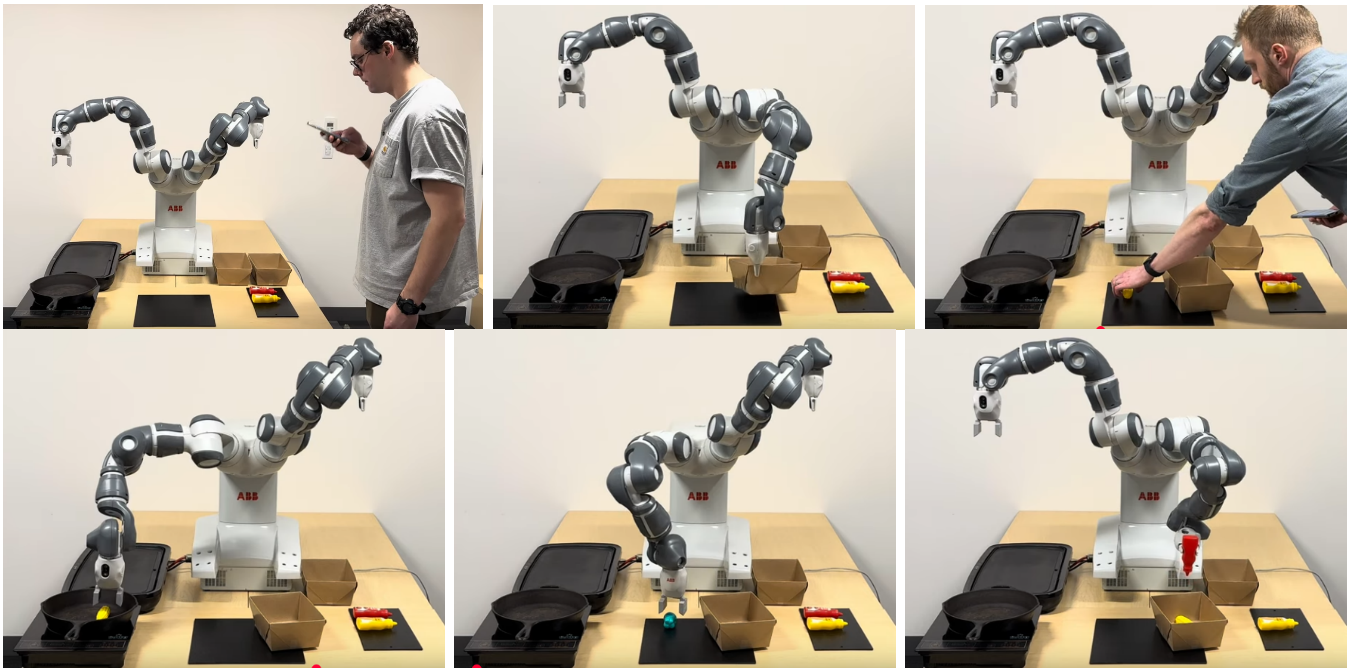

Fig. 2. Different phases of the the Yumi performing the southwest bowl assembly. Top left to right: the initial task setup and instruction, grabbing the container, and getting the first ingredient. Bottom left to right: cooking the corn, grabbing the pepper, adding ketchup.

users to refer to known recipes like the previously learned "southwest bowl" and only teach the modifications:[5]

```
H: Define new item by analogy Puerto
   Rican bowl.
R: Okay. What item is it like?
H: Southwest bowl.
R: Okay. What are the differences?
H: Replace 'right arm saute the carrot for
   30 seconds' with 'right arm saute the
   chicken for 45 seconds'.
R: Okay are there any more differences?
H: Remove the bell pepper.
R: Okay are there any more differences?
H: Add 'get a plantain in the serving box'
   after 'get the chicken in the serving box'.
R: Sorry, I don't know what 'plantain' means.
H: Suspend current task.
R: Okay.
```

The instructor pauses the instruction to teach the robot about plantains.

```
H: Now define new ingredient 'plantain'.
R: Okay. Where is it located?
H: Pose pantry.
R: Okay. What type of ingredient is it?
H: Topping.
R: Okay. What vision job is used to detect it?
H: Detect plantain.
R: Can you show me how to grasp a plantain?
H: Grasp it here.
```

The instructor moves the robot's left gripper to the grasp points and the robot acknowledged with "Okay". Now that the robot knows what a plantain is, the modification instruction can be resumed and finished:

```
H: Resume previous task.
R: Okay. Are there any more differences?
H: Add 'get a plantain in the serving box'
   after 'get the chicken in the serving box'.
R: Okay, are there any more differences?
H: No more differences.
R: Okay.
```

Now that the robot knows about Puerto Rican bowls, it can be prepared:

```
H: Prepare a Puerto Rican bowl.
R: What type of sauce?
H: Aioli sauce.
```

The robot then starts to prepare it based on the instructions it received:

```
R: Could you please bring a chicken from the
   pantry to the prep area?
H: Here it is.
R: Could you please bring a corn from the
   pantry to the prep area?
H: Here it is.
R: Could you please bring a plantain from the
   pantry to the prep area?
H: Here it is.
```

And again, the system announces when the order is complete by saying "Puerto Rican bowl is complete".

VII. Discussion

It is important to point out that the employed DIARC version does not utilize any form of generative AI (e.g., [10])

[5]A video showing the interaction can be found at `https://youtu.be/0MieedVay6k`.

and thus does not depend on expensive cloud-based resources but can rather run in a distributed fashion on the robots themselves. In addition to saving massive amounts of energy and computational resources and thus having an environmental footprint several orders of magnitude less than what current reasonably-sized VLAs would require, there are several other advantages to our approach: (1) no large-scale pre-training is needed, nor is it necessary to collect large amounts of data to train long action sequences (only training of behavior primitives is necessary, which can also be done by moving the robot arm along a motion trajectory using kinesthetic teaching, and the planner will then utilize them to generate long action sequences); (2) while the language is restricted to the domain of operation and allows for only small variations in the way instructions are expressed, once the NLU system has been adapted to it, the behavior is guaranteed when the system is given an instruction. That is, if the system is able to generate the appropriate semantic representation for it, we can guarantee (formally) that the corresponding actions will be taken as instructed — different from foundation models where no such guarantees are possible; this is particularly important with systems interacting with humans in common physical spaces; (3) any number of heterogeneous robots can be added to the system without requiring any additional training; the system can determine what actions each robots can perform and assign robots to tasks accordingly; (4) by having introspective access to all system knowledge, from initially given knowledge to acquired knowledge, the robot system is completely "transparent and explainable" in that human operators can at any time query the system about its knowledge, goals, and plans; and (5) by allowing modifications to any aspect of the system — from the types and numbers of robots, the instructions they understand, and the skills and tasks they learned — any aspect of the system can be quickly adapted to new task settings, again without the need to collect large mounts of data.

On the flip side, the language aspects of the system are limited to task-based dialogues where human instructors need to break down more complex tasks into actions the robots are able to perform (i.e., the executable action sequences are fully determined by the system's primitives). The knowledge the system can acquire is also restricted to what can be expressed in the formal semantics that are used to define the task domain (as opposed to the more general knowledge representations of LLMs in natural language), and for actual deployments, the implementations of the robots' primitive actions will have to be improved to make them more robust. However, the impact of these restrictions pales compared to the overhead current foundation models incur, not to mention the fact that performance on VLAs on robots does not yet even reach the performance of other past approaches. We believe that the proposed system presents a very viable alternative to current power and resource-hungry approaches based on generative AI, and the demonstration of the food preparation task shows that it is ready for deployment.

VIII. Conclusion

In this paper we presented the cognitive robotic DIARC architecture for multi-human, multi-robot, mixed-initiative teams where multiple humans have to perform tasks together with multiple, possibly heterogeneous, robots. While we were not able to demonstrate all teaming capabilities of the system due to lack of space, we did show its ability to quickly learn new tasks from instructions in a practical food preparation task that requires planning and execution with multiple robots (in our case two independent arms mounted on a single platform) that could then perform the task immediately, interacting with humans when help was needed. The proposed system could be immediately applied in fast food restaurants, and beyond in any other teaming setting only requiring definitions of the robot's sensing and action primitives together with basic task-specific concepts. Critically, the DIARC architecture runs entirely on edge systems without the need for foundation models and does not need any expensive power-demanding computational infrastructure, thus allowing for a cost-effective way to operate the multi-robot system in practical settings where cloud-based access might not be possible or financially viable.

References

[1] R. Paleja, M. Munje, K. Chang, R. Jensen, and M. Gombolay, "Designs for enabling collaboration in human-machine teaming via interactive and explainable systems," *Advances in Neural Information Processing Systems*, vol. 37, pp. 64 942–64 969, 2025.

[2] A. S. George and T. Baskar, "The rise of intelligent automation: How advances in robotics and ai are reshaping industries," *Partners Universal Innovative Research Publication*, vol. 3, no. 1, pp. 1–8, 2025.

[3] M. Scheutz, S. DeLoach, and J. Adams, "A framework for developing and using shared mental models in human-agent teams," *Journal of Cognitive Engineering and Decision Making*, vol. 11, no. 3, pp. 203–224, 2017.

[4] S. Goel, P. Lymperopoulos, R. Thielstrom, E. Krause, P. Feeney, P. Lorang, S. Schneider, Y. Wei, E. Kildebeck, S. Goss, M. C. Hughes, L. Liu, J. Sinapov, and M. Scheutz, "A neurosymbolic cognitive architecture framework for handling novelties in open worlds," *Artificial Intelligence*, vol. 331, p. 104111, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000437022400047X

[5] M. Scheutz, B. Oosterveld, J. Peterson, E. Wyss, and E. Krause, "A multi-robot architecture framework for effective robot teammates in mixed-initiative teams," in *Proceedings of Technical Advances in Human-Robot Interaction (THARI'24)*, 2024.

[6] M. Scheutz, T. Williams, E. Krause, B. Oosterveld, V. Sarathy, and T. Frasca, "An overview of the distributed integrated affect and reflection cognitive diarc architecture," in *Cognitive Architectures*, ser. Intelligent Systems, Control and Automation: Science and Engineering, M. I. A. Ferreira, J. S. Sequeira, and R. Ventura, Eds. Springer International Publishing, 2019.

[7] P. Schermerhorn, J. Kramer, T. Brick, D. Anderson, A. Dingler, and M. Scheutz, "Diarc: A testbed for natural human-robot interactions," in *Proceedings of AAAI 2006 Mobile Robot Workshop*, 2006.

[8] M. Scheutz, "ADE - steps towards a distributed development and runtime environment for complex robotic agent architectures," *Applied Artificial Intelligence*, vol. 20, no. 4-5, 2006.

[9] ——, ""teach one, teach all" – the explosive combination of instructible robots connected via cyber systems," in *IEEE Cyber*, 2014.

[10] B. R. Brown, "Human–machine teaming using large language models," in *Interdependent Human-Machine Teams*. Elsevier, 2025, pp. 41–66.