CASPER: Cognitive Architecture for Social Perception and Engagement in Robots

VINANZI, Samuele and CANGELOSI, Angelo

International Journal of Social Robotics
https://doi.org/10.1007/s12369-024-01116-2

CASPER: Cognitive Architecture for Social Perception and Engagement in Robots

Samuele Vinanzi[1] · **Angelo Cangelosi**[2]

Accepted: 8 February 2024

Abstract
Our world is being increasingly pervaded by intelligent robots with varying degrees of autonomy. To seamlessly integrate themselves in our society, these machines should possess the ability to navigate the complexities of our daily routines even in the absence of a human's direct input. In other words, we want these robots to understand the intentions of their partners with the purpose of predicting the best way to help them. In this paper, we present the initial iteration of cognitive architecture for social perception and engagement in robots: a symbolic cognitive architecture that uses qualitative spatial reasoning to anticipate the pursued goal of another agent and to calculate the best collaborative behavior. This is performed through an ensemble of parallel processes that model a low-level action recognition and a high-level goal understanding, both of which are formally verified. We have tested this architecture in a simulated kitchen environment and the results we have collected show that the robot is able to both recognize an ongoing goal and to properly collaborate towards its achievement. This demonstrates a new use of qualitative spatial relations applied to the problem of intention reading in the domain of human–robot interaction.

Keywords Cognitive human–robot interaction · Cognitive architectures · Cooperating robots · Social human–robot interaction · Intention reading · Artificial intelligence

1 Introduction

Autonomous robots are increasingly present in our everyday life. Once limited to research laboratories and industrial settings, they now frequently inhabit our living spaces and interact with us during our day. This new generation of intelligent machines, categorized under the umbrella term of "social robotics", is expected to navigate a complex and uncertain landscape made of human beliefs, desires, intentions and social norms. Additionally, it is desirable for these agents to be able to act autonomously, that means without direct input from their human partners.

In order to integrate these robots into our society, it is paramount for them to be endowed with the same set of cognitive and mental skills that regulate the way in which we, as people, interact with other agents. One of the most fundamental of such skills is known as "intention reading" and represents the capacity to understand the implicit goal that is driving the actions of another agent [1]. By making appropriate use of this ability, we can allow a social robot to use its observations of another agent to extrapolate their underlying goal, reconstruct their expected future actions and, finally, determine how and when to enact collaborative behavior.

This paper presents our first, inaugural version of CASPER (cognitive architecture for social perception and engagement in robots), a platform-independent cognitive architecture that uses a mixture of symbolic and data-driven artificial intelligence methodologies to perform intention reading and collaboration in a human–robot interaction (HRI) scenario. "Social perception" refers to the act of identifying and using social cues to make judgments about others, while with the term "engagement" we highlight the system's ability of translating this knowledge into practical involvement and interaction.

This system lays its foundations on the use of qualitative spatial relations (QSRs) that describe how the observed partner moves inside the environment with respect to the Objects of Interest (OOIs). CASPER analyzes the temporal evolution of these descriptors to predict future actions, which are

✉ Samuele Vinanzi
s.vinanzi@shu.ac.uk

[1] Department of Computing, Sheffield Hallam University, Sheffield, UK

[2] Manchester Centre for Robotics and AI, The University of Manchester, Manchester, UK

Published online: 14 March 2024

subsequently validated by a knowledge-base system and processed by a probabilistic plan recognizer. This data-driven approach to recognition is complemented by a top-down decision-making strategy to generate assistive activity.

Our contributions to the field include the following:

- A new cognitive architecture that implements intention reading and collaborative behavior capabilities for human–robot interaction, the need of which arises from the scarcity of such models in the current literature [2].
- To the best of our knowledge, CASPER represents one of the first attempts to introduce the use of QSR descriptors to perform efficient and easily generalizable intention reading for embodied robots. This paper presents a proof of concept on the untapped potential of these mathematical tools and tries to promote their use in future cognitive architectures for action and goal recognition.
- On a technical level, we present a collection of novel algorithms for social perception and reasoning that take inspiration from well-established psychology and cognitive science principles.
- We provide the first case study for this cognitive architecture: a demonstration of its possible implementation to solve a collaborative task based in a kitchen environment.

Our long-term goal is to implement CASPER into an heterogeneous multi-agent teaming scenario, where a team of distributed agents, both humans and robots of different kind, are engaged in joint action to achieve a common goal. In order to do so, this paper builds the foundations of this cognitive architecture and presents a first case study involving a dyadic interaction that takes place inside a kitchen. Within this specific application of CASPER, the robot is expected to observe its partner, identify the goal that is driving their actions and calculate the best way to assist with their task. Our experiments carried out in simulation (Fig. 1) show that the robot using this cognitive architecture is able not only to accurately predict the partner's goals before they are achieved, but also to formulate appropriate collaborative decision-making. This allowed the human to silently and implicitly delegate part of the task to their artificial companion.

Fig. 1 The simulated robot equipped with CASPER observes the actions of another agent (in this case, a simulated human) in order to predict their goal and the best collaborative plan

The organization of the paper is as follows: Sect. 2 offers a general background on artificial cognitive architectures and intention reading, other than introducing the notion of QSRs. Section 3 discusses CASPER's design specifications and algorithmic details. Section 4 covers the implementation of the general CASPER architecture to the specific example of the simulated kitchen environment. Section 5 discusses the performance of the system on the selected case study. Finally, Sect. 6 concludes and highlights possible future directions.

2 Previous Work

2.1 Cognitive Architectures for Robots

Cognitive Robotics is a discipline that lies at the intersection of robotics and cognitive science, which is the scientific study of the mind and its processes such as perception, attention, anticipation, planning, memory, learning, and reasoning. It has been defined as "the field that combines insights and methods from artificial intelligence, as well as cognitive and biological sciences, to robotics" [3]. This definition highlights the interdisciplinary nature of this approach, which takes inputs from linguistics, psychology, neuroscience, philosophy, computer science and anthropology. Its aim is to create intelligent robots which are endowed with the same set of mental skills as a human being.

Cognitive science views the mind as an information processor and studies the operations through which perceptual stimulus are combined to obtain higher-level mental functions [4]. These principles are easily transferable to an embodied robotic platform that can implement the same functions despite the difference in the underlying structure (a brain versus a computer). This is done by designing what is known as an "artificial cognitive architecture": a computational system which instantiates one or more cognitive theories using artificial intelligence methodologies in an attempt to model the human mind.

A recent review has estimated the existence of around 300 cognitive architectures in the current literature [2]. The vast majority of them specialize on modeling particular aspects of cognition such as attention [5], emotion [6] or problem solving [7], while only a fraction aims to achieve Artificial General Intelligence. The latter case includes some of the most famous architectures in the current literature, such as ACT-R [8], Soar [9], LIDA [10] and NARS [11]. These are all implemented as general frameworks that can be deployed to specific use cases, including applications in robotics [12]. For example, ACT-R is written as a Common Lisp interpreter and its applications come in the form of scripts in the ACT-R

language. Despite their purpose, every cognitive architectures usually models one or more aspects of cognition such as perception, attention, action selection, memory, learning and reasoning [2].

CASPER belongs to the category of more specialized architectures and focuses on modeling specifically human–robot collaboration (HRC) mental capabilities and, in particular, intention reading. Other cognitive systems that belong to the same class make use of different techniques such as unsupervised clustering of human postures [1] and artificial mirror neuron networks [13].

2.2 Human–Robot Collaboration

The human species' success is ascribable to its ability to collaborate with others to obtain otherwise inaccessible goals. This instinct towards cooperation is shared by our close relatives in the animal kingdom, which however use it as a means to achieve a purpose, rather than being intrinsically motivated in pursuing it [14]. Given the importance that collaboration has for humans, it seems natural to try and transpose this skill to the autonomous intelligent machines that we are designing as everyday companions.

HRC is a branch of HRI which studies the best ways to ensure a safe and effective interaction between humans and robots engaged in joint tasks with common goals: we call this a "team". The vast majority of works in this field's literature focus on industrial settings, where robotic arms such as Sawyer or Kuka robots offer assistance in some kind of assembly task [15–17]. Many of these studies deal with the scheduling and subdivision of tasks between the two agents. Hoffman et al. [18] argue that collaborative robots (or "cobots") should possess communication mechanisms in order to both understand humans and to inform them about their own goals, and in so doing maintain a set of shared beliefs which support the execution of a joint plan. In fact, many HRC models focus on direct verbal cooperation and implement dialogue managers [19–21]. More recently, a number of studies have adopted Machine Learning methodologies (sometimes encased in artificial cognitive architectures) to learn and modulate the robot's response to human tasks [22].

2.3 Intention Reading

On a psychological point of view, one of the main cognitive skills that enable social and collaborative attitudes in humans is known as "intention reading" [23]: this is the ability to understand the driving goal of another agent by the observation of their physical clues. This is possible because we don't understand the behavior of others as a series of unrelated motions through space, but rather as sets of goal-directed actions [24]. Intention reading is a fundamental skill to implement in a collaborative intelligence because an agent has to first understand what their partner's goal is before knowing how to offer its assistance.

Balwdin et al. [25] state that humans process continuous actions as streams of hierarchical relations that link low-level intentions (such as grasping a plate, bringing it to the sink and opening the tap) to high-level intentions (to wash the dishes or clean the kitchen). Additionally, they state that adults more reliably identify the higer-level goals based on some actions that are understood to be more crucial than others (for example, they point to the fact that scrubbing a plate is a stronger signal for the intention to wash the dishes than the equally necessary but less central action of turning on the water). Some of the most important social cues that are used for this purpose are biological motion and gaze direction [26].

On a computational perspective, there have been many attempts to develop intention reading models for HRI. A notorious experiment by Dominey and Warneken [27] investigates the shared intentionality in a turn-taking game between a human and a robotic arm, where the artificial agent would build a representation of the shared plan and subdivide the actions between itself and its partner. Duarte et al. [28] perform action anticipation exploiting several social cues such as saccadic eye movement, gaze directing and arm movements processed through a Gaussian Mixture Model. Bien et al. [29] developed a system that analyzes posture and movements in elderly people and tries to decode the inner intentions they are driven by. Other relevant researches involve the use of dynamic Bayesian networks [30], self-organizing maps [31], first order logic [32] or the imitation of the biological mirror neuron system [33]. An interesting approach has been adopted by Granada et al. [34], who divide the intention reading task on a low-level action recognition paired with a high-level goal understanding and execute the task combining a convolutional neural network with a symbolic plan recognizer.

The high-level cognitive process of intention reading often relies on additional abilities. One such skill is movement classification, involving the description of human actions using primitive elements. Perceptual data, collected through specialized sensors [35] or vision techniques [1], can be analyzed using models such as Hidden Markov Models, Dynamic Bayesian Networks, and grammars [36]. Once an intention is recognized, plan recognition techniques using AND/OR trees [37] or probabilistic context-free grammars [38] can be used to infer the higher-level objective.

Previous works by the authors [1, 39, 40] have explored the use of social cues and psychological theories to develop intention reading capabilities in humanoid robots. In these works, we analyzed kinematic signals (body posture, head gaze) to infer the human's current goal using clustering algorithms paired with probabilistic modeling. Our interest is shifting from dyadic interactions to heterogeneous multi-

agent environments in which the Team Goal is shared among sub-teams distributed along a structured world. For this reason, we felt the need to adopt a higher-level approach and shift our focus from subtle kinematic signals to qualitative relations between the entities in the environment. This work lays the foundations of CASPER, which will be in the future applied to a complex scenario like the one we have just described.

2.4 Qualitative Spatial Relations

Most mathematical and engineering modeling of phenomena rely on *quantitative* representations: measurements made on standard units that define physical properties. This is not how human commonsense generally works: in order to process information in a timely and efficient manner, we tend to discard unnecessary details, reducing the concept of space into coarse categories. For example, in our everyday life we would say that something is "far away" without specifying the exact number of meters. This is an example of *qualitative* spatial reasoning.

QSRs are tools that allow commonsense reasoning about space and time using qualitative relations for different spatial aspects such as topology, direction and position [41]. In other words, QSRs enable an agent to reason about actions such as "the human is approaching the fridge" without having to maintain precise metric information on the positions of both the actor and the reference point (in this example, the human and the fridge). Such a representation does not depend on factors such as the starting positions of the actor and reference, the speed of movement or its exact trajectory and so it is easily generalizable [42].

QSRs have been used for a broad range of artificial intelligence applications, such as human activity learning [42] and monitoring [43], language learning [44], imitation learning [45] and even to encode spatial structure features for artificial neural networks [46].

The use of QSRs for intention reading purposes is, to the best of our knowledge, novel. Many papers that make use of them focus on action recognition, which is an important but non exhaustive step in the process of understanding the underlying goal that is driving an agent's behavior. One of the main scientific contributions of this paper is to demonstrate that QSRs can be effectively used in this domain, since they are more prone to generalization than the features usually employed by intention reading models (such as motion trajectories or posture keypoints).

3 Proposed Method

The purpose of our work is the development of an artificial cognitive architecture that will allow an autonomous social robot to observe the actions of a partner (be it a human or a humanoid), understand their underlying goal and collaborate on the ongoing task. Our design choices are inspired by the low- and high-level subdivision of actions and goals theorized by psychologists and use biologically plausible inputs [26]. We allow this model to generate an appropriate assistive response by leveraging on the prediction of the current goal, the observed actions and the incomplete part of the plan. The overall system is depicted in Fig. 2. The architecture is composed of several parallel processes which gather, elaborate and share data between them.

3.1 Goal and Intention Representation

Before giving a detailed description of each component showed in Fig. 2, it is worth explaining how this cognitive architecture represents intentions. Within CASPER, every goal is described by a Plan: a sequence of events that have to be executed in order for the task to be accomplished. Figure 3 describes this structure as a non-binary tree. The main *goal* is formed by a collection of *sub-goals*, each of which is achievable by a sequence of *actions*. The latter, in turn, are composed by a set of *movements*. The nodes of the Plan are temporally ordered from left to right.

The nodes represented as ellipses in Fig. 3 are the abstract, conceptual representations of the plan and fall under the domain of the High-Level module (Sect. 3.5), while the ones drawn as rectangles are the ones that can be directly observed by the Low-Level module (Sect. 3.4). The hierarchical organization of goals is inspired by cognitive science [4].

This data structure is used both when reading the partner's intention (bottom-up, data-driven inference) and when generating an appropriate collaborative behavior for the robot (top-down, conceptually-driven inference).

3.2 Environment and Robot Control

This architecture assumes the existence of a robot r which is observing another agent a performing actions within an environment E. The latter will contain a set $O = \{o_1, o_2, ..., o_n\}$ of OOIs which can be interacted with. For example, possible OOIs in a kitchen would be pieces of cutlery and food items, but not structural elements such as walls and floors.

We have designed this cognitive architecture to be platform-independent. This means that any robot with basic vision capabilities can be programmed to interact with CASPER. Of course, the physical limitations of the chosen robot define how it will be able to interact with the world and assist its partner, but this information can be easily encapsulated in the knowledge base (see Sect. 3.6).

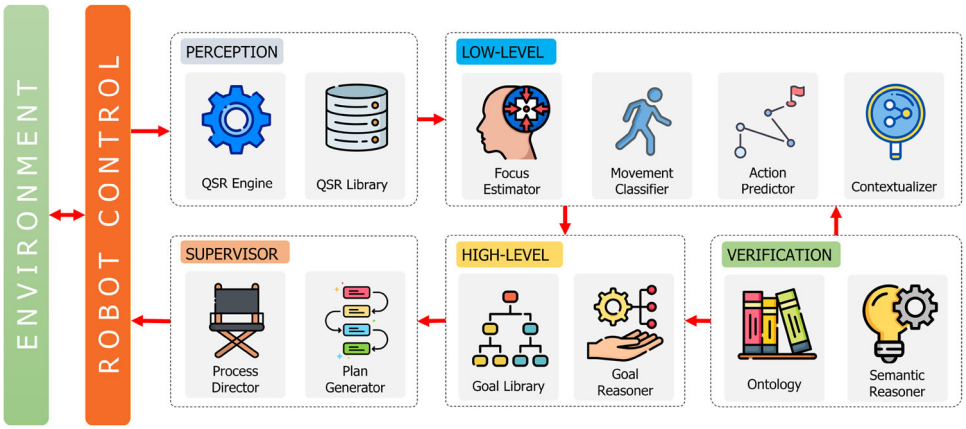


Fig. 2 Overview of the proposed system. CASPER is composed of several parallel processes that interact with each other in a joint effort to decode an agent's intention and to formulate an appropriate response. The Perception module transforms visual observations into QSRs. The latter are used by the Low-Level process to predict the actions that are being performed in the environment and passes this information to the High-Level component, which tries to match them against the plan library to infer the pursued goal. A knowledge base, enveloped in the Verification module, ensures the step-by-step soundness of these predictions. Finally, the Supervisor coordinates all the other processes, collects the results and composes a collaborative plan that will be executed by the robot

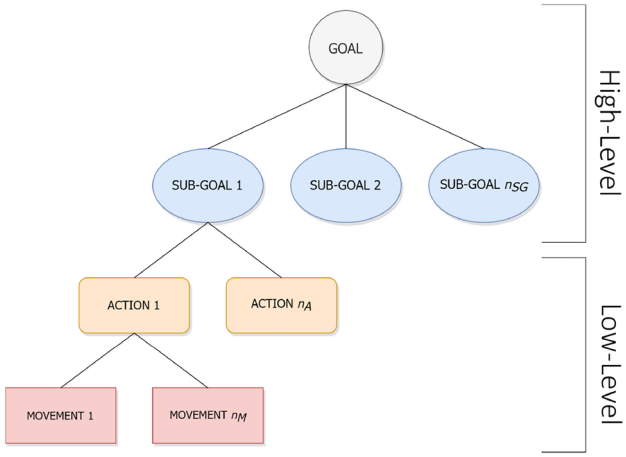


Fig. 3 A plan in CASPER. Each goal is formed by a temporally ordered set of sub-elements with varying levels of abstraction. This structure is used for both intention reading and collaborative behavior generation

3.3 Perception

The Perception module interacts directly with the Robot Control system and is used to produce a qualitative description of the relations between the observed agent and the OOIs in the environment. The agent will periodically update this component by transmitting a World State: a dictionary that, at every timestep, records the absolute coordinates of each OOIs calculated from the robot's visual sensors (such as RGB or RGBD cameras). This World State is then processed by the QSR Engine to obtain the qualitative spatial descriptors of the scene at the current timestep. For our purposes, we chose to use QSRlib [47], an open-source software library designed to calculate QSRs from a scene description. Employing an established library like QSRLib eliminates the need to implement the mathematical formulations for these metrics from scratch.

The QSRs we calculate from our World State are the following:

- Qualitative Distance Calculus (QDC) [48].
- Qualitative Trajectory Calculus (QTC) [49].
- Moving or Stationary (MOS).
- Holding Object (HOLD).

QDC: defines the qualitative Euclidean distance between two entities in the scene, which in our case are a and $o_i \in O$. On an intuitive level, this QSR describes how close the agent is to the OOIs under consideration. The thresholds are parameterized within QSRlib and are defined as: 'touch' [0–0.6m], 'near' (0.6–2 m], 'medium' (2–3 m], 'far' (3–5 m] and 'ignore' for distances greater that 5 m.

QTC: represents the relative motion between a set of moving point objects having a free trajectory in an n-dimensional space. The current literature contains several variations of this descriptor [50], out of which we chose QTC$_{B11}$. The latter involves two points (which for our purposes will be a and o_i) and makes use of the Euclidean distance calculated on the reference line that connects them. Because of the nature of this QSR, it can only be calculated over two distinct timesteps. The results of this calculation can be either: a is stationary with respect to o_i (represented by the symbol 0), a is moving towards o_i $(-)$ or a is moving away from o_i $(+)$.

MOS: a unary QSR that describes whether the entity is in motion or stationary between two different timesteps.

HOLD: this unary descriptor indicates whether the agent a is holding an object in one of their hands or not.

After having produced a QSR description of the current timestep, the Perception module stores it in a local library, which acts as the system's sensory memory. The latter contains all the time-ordered QSRs calculated since the beginning of the activity and it can be accessed on request by other processes.

3.4 Low-Level Action Recognition

This component of the cognitive architecture is in charge of predicting the first elements of the data-driven inference as shown in Fig. 3: movements and actions. The main idea behind this module is to incrementally aggregate and refine data: a set of QSRs will be classified as a movement and a set of movements will define an action.

3.4.1 Focus Estimation

If there are n OOIs in the environment, the Low-Level would be able to perform n distinct classifications, one for each OOI. This means that, at every timestep, CASPER would be able to identify n different movements. In reality, we know that the partner is an intentional agent which is performing one movement at a time, directed towards a specific entity in the scene which we call the "target".

The challenge, then, is to identify the target using perceptual information from the observed scene. Traditional techniques utilize gaze modeling [51], but this information might be inaccessible to the robot due to its relative position in relation to the human. Given the inherent uncertainty of this task, it seems natural to solve it through the use of probabilistic models. Our algorithm assigns a score S to each OOI based on the following equation, which envelops the use of motion and gaze as perceptual inputs to the intention reading cognitive functions theorized by Tomasello et al. [26]:

$$S(o_i) = \frac{w_{QDC} \cdot QDC(o_i) + w_{QTC} \cdot QTC(o_i)}{1 + \theta} \quad (1)$$

Table 1 Encodings for the QSRs used in the focus estimator

QDC	Encoding
Touch	0.5
Near	0.25
Medium	0.125
Far	0
QTC	**Encoding**
0	0.5
−	0.25
+	0

for $o_i \in O$. In the above equation, $QDC(o_i)$ and $QTC(o_i)$ represent the QDC and QTC QSRs calculated on OOI o_i. These are categorical variables, so they need to be encoded into numerical values using the conversion shown in Table 1. w_{QDC} and w_{QTC} are the positive weights that we assign to these components, with $w_{QDC} + w_{QTC} = 1$.

Given an uniform weight distribution, the numerator in Eq. 1 is maximized when the agent is maintaining touching distance with the object. On the contrary, it is minimized when the agent is at a far distance, walking away from it. This value is scaled by θ, which represents the angle between the agent's heading and the reference line connecting them with the OOI. Intuitively, the denominator penalizes OOIs which are not in the field of view of the agent.

After calculating the attention score for every OOI, these are normalized into probabilities. To win the competition and be elected as the target, an element must both possess the maximum score and surpass the threshold $\tau = 0.5$. To eliminate any possible noise in the prediction that would affect the processing chain, this item is not forwarded as it is, instead it is inserted into a sliding window of size $w = 4$ which allows the system to select the target as a measure of central tendency. Any OOI that at any time occupies the majority of the sliding window slots is declared to be the current target of the observed agent's focus.

The Focus Estimator keeps also track of the second-highest scoring OOI, processing it independently using the same procedure described above. This element, when it exists, is assumed to be the "destination", which will be later used to contextualize the agent's action. For example, if the agent is transporting an empty glass towards a bottle, then they are probably going to fill it and have a drink, conversely if the destination is the sink, then they will likely going to wash it.

In case a tie occurs between the OOIs, if one of them was previously declared as a target it will maintain its status, while the other one will be regarded as the destination.

3.4.2 Movement Classification

Once the Focus Estimator has identified the partner's target, it is possible to proceed with the movement prediction with respect to the inferred OOI. For this purpose, we make use of a symbolic data-driven model: specifically, a Decision Tree [52]. We have chosen this model because it fits well for our purpose, that is to form a mapping from a set of QSRs to a domain of discrete movements, each representing a motion that the observed agent is performing in a single timestep.

A graphical representation of this process can be viewed in Fig. 4.

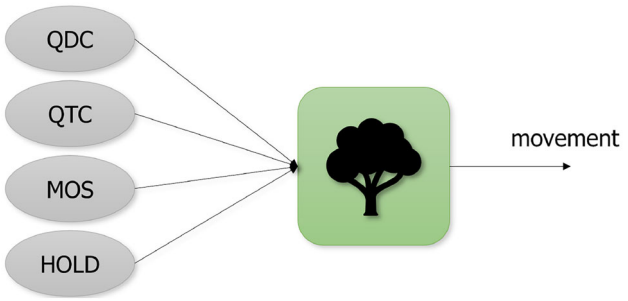


Fig. 4 The Decision Tree maps a set of QSRs into a movement

3.4.3 Action Prediction

Through the procedures described in the previous sections, CASPER is able to determine the movement performed by the agent at each timestep, providing the robot with an instantaneous information of what is happening in a single unit of time. To fully understand the behavior of the partner, however, we need to analyze the temporal evolution of these movements: we call this an "action".

To represent the composition of each action, we use a Markov-chain Finite State Machine (FSM) [53]. This model describes a process in which the transition to a state at time $t+1$ is probabilistic rather than deterministic and depends only on the state at time t. Then, we combine these FSMs in an ensemble which is used to classify a sequence of observations. Every time the Movement Classifier generates new data, this is initially filtered such that only transitions between different states are considered. This technique, already implemented in other activity recognition and prediction models [1, 54], allows the system to be time-invariant: this means that the speed at which the action is performed does not influence its representation.

Every time a new filtered observation is detected, it is queued up with the previous ones. The ensemble samples each of its FSMs using the initial observation as the first state of the chain. Thereafter, it calculates the similarity between the ordered sequence of observations and the generated samples using the Ratcliff-Obershelp Pattern Recognition algorithm [55]. Each FSM is assigned a score in [0, 1] based on this metric. An action is predicted when there is a clear winner between the models, but only if the score surpasses a certain threshold: this allows the system to be more robust to transient effects that will create noise in the inference.

Actions might be ambiguous and require contextualization based on the location within the environment in which they are performed. When this is the case, we have opted to use a simple lookup table in which we use information from the destination (see Sect. 3.4.1) to disambiguate actions.

3.5 High-Level Goal Prediction

The purpose of this component is to form a computational representation of the plan structure described in Fig. 3 on which it is possible to execute inference and reasoning. We model this data structure as a non-binary tree where the root represents the goal and each terminal node is a possible action (derived from sequences of movements as described in Sect. 3.4).

The Plan Library L is then defined as:

$$L = \{\Sigma, NT, G, T\} \tag{2}$$

Where Σ is a set of terminal symbols that represent the observable actions, NT are the non-terminal symbols which stand as sub-goals, G is the set of goals and T are the trees that describe the ordered production rules that compose the plans.

During the intention reading process, the Low-Level module will produce a serialized set of observed actions $\hat{\sigma} = \{\hat{\sigma}_1, ..., \hat{\sigma}_n\} \in \Sigma$ that have to be matched against the available plans T in the Plan Library in order to infer which goal $g \in G$ is driving the observable actions of the agent. We work on the assumption that $\hat{\sigma}$ is temporally ordered, which means that $\hat{\sigma}_1$ happens and is observed before $\hat{\sigma}_2$ occurs.

The intuition behind our design is the following: using Occam's razor principle, the plan that better describes the data is the one that more simply fits the observations and that leaves less gaps in the explanation (intended as nodes that should have been observed but are not present in $\hat{\sigma}$).

The probability that goal g is generating the observations $\hat{\sigma}$ is:

$$P(g|\hat{\sigma}) = \eta \cdot s(g) \tag{3}$$

Where the score s is defined as a function on the number of observed nodes and the missed ones:

$$s(g) = observed \cdot (1 - missed) \tag{4}$$

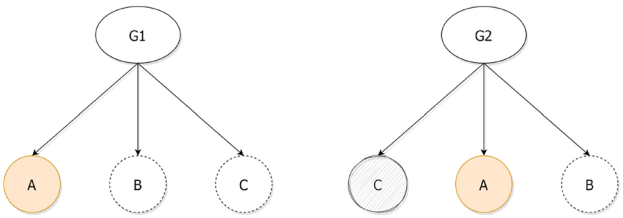


Fig. 5 A visual demonstration of the Plan Library's scoring system. These trees represent two plans for two distinct goals with a single observation $\hat{\sigma}_1 = A$. The non-root nodes are drawn differently based on their status: filled if observed, dashed if unobserved and textured if missed. In this example, $P(G1|\hat{\sigma}) = 0.59$ and $P(G2|\hat{\sigma}) = 0.39$, so $G1$ is considered the best explanation

And η is a normalization factor calculated as:

$$\eta = \frac{1}{\sum_{g \in G} s(g)} \tag{5}$$

Equation 4 penalizes the explanations that contain missed nodes, which are nodes that should be present in the description but don't appear in $\hat{\sigma}$. This process is explained graphically in Fig. 5.

Algorithm 1: Explanations from partial observations

```
Input: A plan library L, a set of observations σ̂
Output: A set of ranked explanations P
Initialize P with the unmarked plans T in L
foreach σ̂_i ∈ σ̂ do
    Initialize P′ to empty
    while P is not empty do
        Pop p from P
        foreach unobserved node n in p named σ̂_i do
            Generate a copy p′ of p
            Mark n as observed in p′
            if there are any unobserved nodes on the left of n in p′
            then
                Mark them as missed
            end
            Insert p′ in P′
        end
    end
    Insert P′ in P
end
Calculate the score for each plan;
return The generated explanations P, ordered by score
```

The procedure to derive the best explanation from a set of observations is described in Algorithm 1. The process dynamically generates a set of explanations, each of which accounts for every possible interpretation of the observed symbols. This means that Algorithm 1 is able to deal with missed observations, where the robot might for any reason not record some of the actions performed by the partner. The explanation with the highest score at the end of the computation is chosen as the inferred intention.

3.6 Real-time Verification

The Verification module represents the robot's cognitive common sense and is responsible for the correctness of the predictions formulated step-by-step by both the Low-Level and the High-Level. In its essence, it serves the purpose of filtering out predictions that might arise at any level of abstraction which do not constitute valid statements on the state of the world. For example, an action prediction such as "the human picks up the table and places it in the oven" might be a possible statement generated by the Low-Level, but it makes no logical sense and must be a product of noise or a transient state.

In order to address the problem, we make use of a knowledge base in the form of an ontology in which we represent the entities of our world and the relations between them. Every time the Low-Level infers an action or the High-Level generates an explanation, these are verified through a Semantic Reasoner and if they are proven invalid they are discarded from the processing pipeline. Additionally, this component is also used by CASPER to decide which actions are assignable to the robot during the final collaborative decision-making. As an example, it would not be possible for the robot to "eat the meal", while on the contrary it would be capable of performing the action "wash the dishes".

On a technical level, CASPER makes use of an OWL2 ontology and a Pellet reasoner [56].

3.7 Collaborative Intelligence

The final element of CASPER is the Supervisor, which is in charge of the coordination of the sub-processes that constitute the cognitive architecture. In particular, it manages the communication between these processes and collects the data produced by the Low-Level and High-Level in order to achieve the final purpose of this architecture: generate a collaborative behavior to help the partner with whatever tasks they are involved with.

To do so, it uses the goal explanation produced by the High-Level to generate an appropriate assistive plan. This involves the robot understanding which actions are yet to be executed and reason about which ones it is able to assist with. From the goal explanation, the Supervisor obtains the "frontier", that is the ordered set of all the unobserved nodes: these are the actions that have yet to be performed in order to achieve the goal. Using the validation procedure explained in Sect. 3.6, it then identifies the longest sequence of actions which the robot itself is able to perform. It then continues observing, waiting for the partner to execute the rest of the plan up to the point where the collaboration will start and in that moment it will send instructions to the Robot Control.

4 Experiments

4.1 Experimental setup

In order to test the effectiveness of CASPER, we have decided to deploy it on a selected case study. In particular, we have developed an experiment involving a human and a TIAGo++ robot interacting inside a kitchen containing 7 OOIs: a water bottle, a canned meal, a box of biscuits, an empty glass, a plate, a set of hobs and a sink. This environment is shown in Fig. 6. This was developed as a virtual environment created using the Webots open-source physics simulator [57].

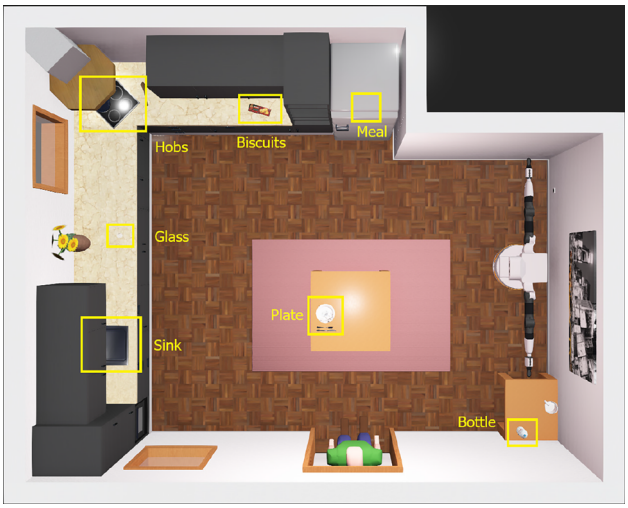


Fig. 6 The experimental kitchen environment rendered in Webots. A TIAGo++ robot is tasked with observing a human performing actions in the scenery. The 7 OOIs are marked and annotated

The robot is instructed to visually find and track the human, collect observations on their actions within the room and process them through CASPER to infer their goal and generate an appropriate collaborative behavior. Both the human and the robot are able to navigate the environment, grasp items and release them.

4.2 Low-Level

The *movements* we have defined for our experiment are the following:

- **STILL**: the agent is fixed in space, neither moving or interacting.
- **WALK**: the agent moves in the environment, holding nothing.
- **TRANSPORT**: same as WALK, but performed while carrying an object.
- **PICK**: the agent collects and item.
- **PLACE**: the agent positions a previously collected object somewhere in the environment.

In order to train the Decision Tree to map a set of QSRs to these movements, we need to generate a dataset. We do this by positioning the human in a random position inside the room and tasking them to stay still for a while, then walk to a random OOI, pick it up, transport it to a random destination, place it and then stand still again. This demonstrates the full range of movements that we wish to learn through our model. The robot, in turn, will observe the scene, calculate the QSRs and associate them to a label which is manually provided by the experimenter. A Decision Tree is a small-data model so we don't require a large amount of training samples: we repeat the previous procedure 10 times, then fit the model with the collected data.

The chosen *actions* for the kitchen setup are:

- **Pick and place**: a PICK movement followed by a TRANSPORT and terminated by a PLACE.
- **Use**: a loop of PICK and PLACE movements.
- **Relocate**: STILL, followed by WALK and another STILL movement.

The FSMs that describe these actions are reported in Fig. 7.The transition probabilities have been designed based on the expected sequence of actions the model aims to rec-

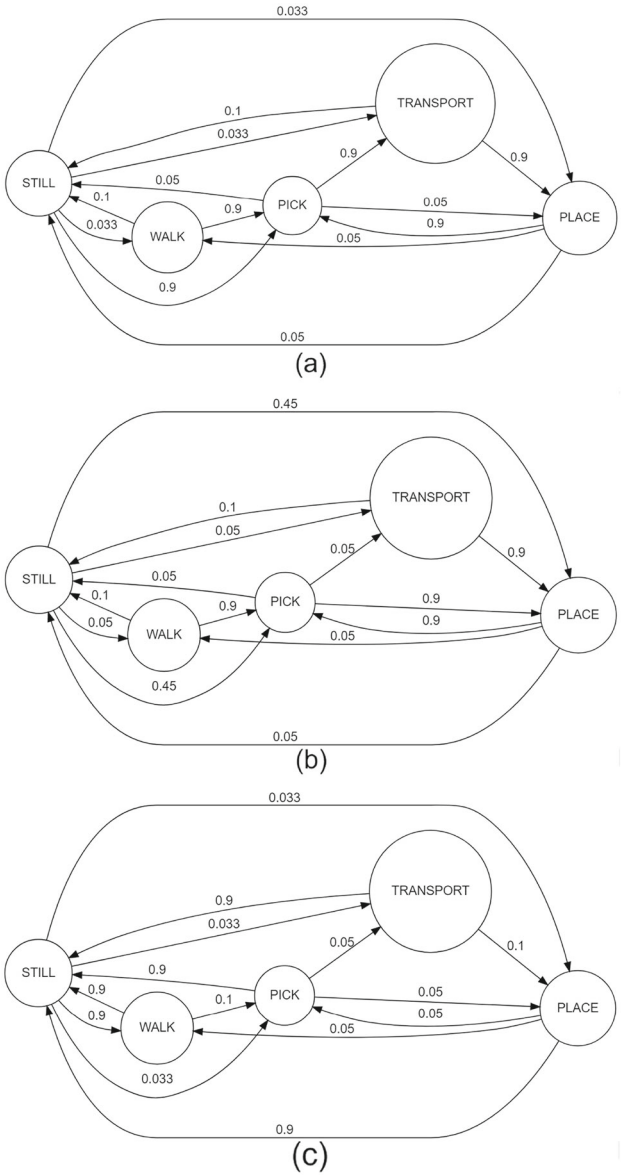


Fig. 7 The FSMs which describe how each action is composed from the primitive movements: **a** Pick and place, **b** use, **c** relocate

Table 2 Contextualization for the action 'Use'

Destination	Contextualized Action
Sink	Wash
Hobs	Cook
Plate	Eat
Glass	Sip

ognize. Additionally, a margin for a small degree of potential error was accounted for, allowing for a certain level of flexibility within subsequent state changes.

There is one action which requires contextualization: Use. This is because the latter can have a different meaning based on the location in which it is performed, or in other words the destination of the action. The lookup table that we use to contextualize it is reported in Table 2.

4.3 High-Level

We define 3 distinct goals for our experiment:

- **Breakfast**: the human will collect the biscuits, bring them to the plate and eat them, then move the plate to the sink and rinse it.
- **Drink**: the human will fetch the bottle of water, bring it to the glass and have a sip, then wash the glass.
- **Lunch**: the human will walk to the fridge and collect the canned meal, place it on the hobs and cook it. Afterwards, they will bring it to the plate, eat and wash the dishes.

The detailed plans for each of these goals are depicted in Fig. 8.

In our plan library, as defined in Sect. 3.5:

$$\Sigma = [PickAndPlace, Wash, Cook, Eat, Sip] \quad (6)$$

$$NT = [PrepareMeal, Warm, Clean] \quad (7)$$

Note that the action Relocate is missing from Eq. 6: this is because our current experiment involves a single room. We have nevertheless implemented this action because of our future development plans for CASPER (see Sect. 6).

4.4 Verification

The ontology which we use to describe the kitchen experiment is reported in Fig. 9. This knowledge base defines each entity in the environment as belonging to one of three macro-groups: Goals, Agents or Objects. Each Agent can be a Human or a Robot, the latter only containing the TIAGo++ robot we are using but potentially expandable to include several kinds of robots grouped by their capabilities (for example, humanoid and non-humanoid). The Objects can

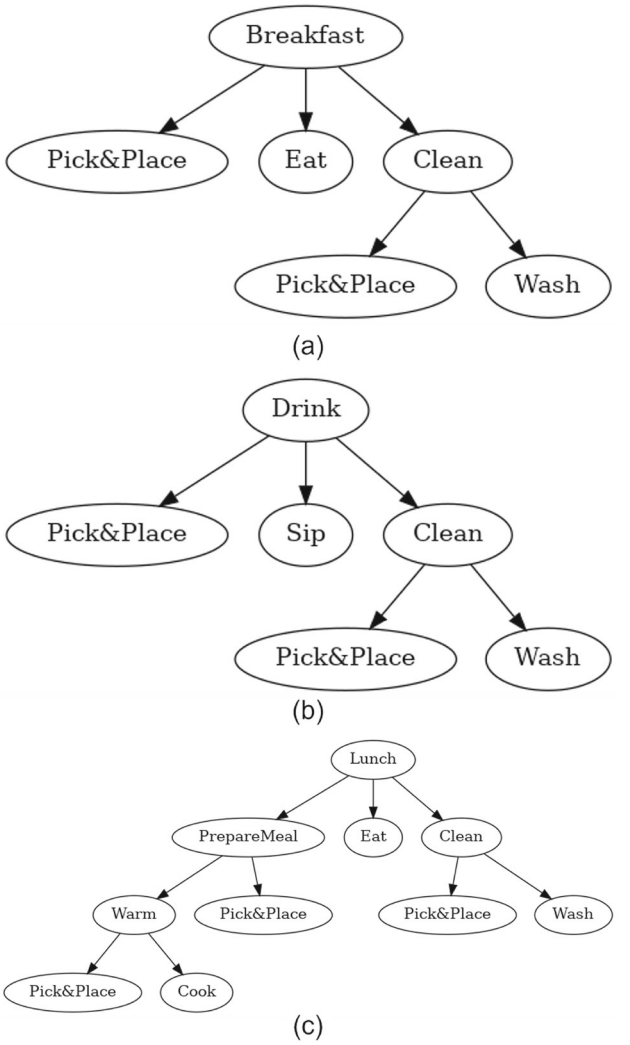


Fig. 8 Detailed plans for each of the goals of the kitchen experiment: **a** Breakfast, **b** Drink, **c** Lunch. In each graph, the root node is the goal, every non-terminal represents a sub-goal and each terminal depicts an action. Movements are not shown for clarity

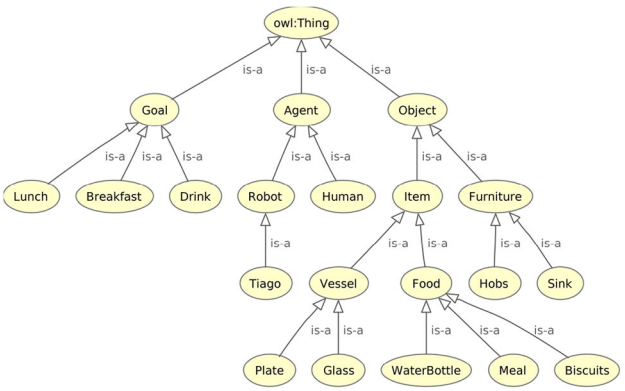


Fig. 9 The ontology used as a knowledge base to perform verification during the kitchen experiment

Table 3 Data properties for the ontological representation of the kitchen experimental environment

Sink	Move	Eat	Drink	Cook	Wash
Hobs					
Plate	✓				✓
Glass	✓		✓		✓
Biscuits	✓	✓			
Meal	✓	✓		✓	
WaterBottle	✓		✓		

be Items or Furniture: the former includes both vessels and food.

Each element is characterized by some properties which define the kind of interactions that are possible with each of them. These are reported in Table 3. For example, the biscuits can be eaten and moved, but not cooked, drinked or washed. At the same time, the ontology defines some object properties, i.e. relations between entities of the knowledge base which impose limitations useful to verify the validity of the statements produced by the Low-Level and the High-Level. For example, the Eat action might only be performed by Humans with a target which is eatable and a destination which is a Vessel. A pair of object properties are defined for each of the actions (including the contextualized ones).

Finally, we use Semantic Web Rule Language (SWRL) [58] definitions to allow the reasoner to perform inferences on incomplete statements. For example, we know that if an Agent is washing an item, then the destination must be a Sink.

5 Results and Discussion

5.1 Focus Evolution in Time

Figure 10 explores the output of the Focus Estimator while observing a human performing the goal Lunch (which, we recall, is executed by collecting the meal from the fridge, cooking it on the hobs, eating it at the table and finally washing the dishes). In particular, each OOI o_i is anno-

Fig. 10 Each OOI is annotated with a graph showing the evolution of the focus estimation probability in time during the execution of the goal 'Lunch'. The arrow on the floor shows the trajectory of the human through the environment

tated with a graph describing the temporal evolution of the assigned normalized probability values $P(o_i)$. Of course, at each timestep: $\sum_{o_i \in O} P(o_i) = 1$.

At the start of the simulation, the farthest OOIs have the lowest chance to be considered as the target of the observed agent's attention. When the human starts moving and turns to their right, the probability for the Bottle increases up to the designed threshold of 0.5, but soon after they face away and the score drops. Thanks to the sliding window, this OOI is not realistically considered as the human's target.

The agent then continues its path to the north: the Plate's probability decreases as it exits their field of view, while the probabilities for the Meal increases steadily. Around timestep 20, the human turns momentarily towards the Biscuits and once again creates a probability spike which does not last long enough to influence the system. Once the Meal is grasped, $QDC(Meal) = Touch$ and $QTC(Meal) = 0$, so the focus estimation for this OOI is high.

Further ahead in time, around timestep 50, the focus is evenly divided between the Meal and the Hobs during the Cook action. The human then turns towards the table: the probability for the Hobs decreases rapidly while the ones for the Plate increase. At this point, around timestep 90, the human finishes approaching the table and starts eating. The Focus Estimation divides more or less equally the probability for both the Meal and the Plate, with a 60/40 split.

Finally, the Plate is brought to the Sink. The focus score for the former is high and eventually evens out with the latter, while the probability value of the Meal drops to 0 very quickly.

Overall, Fig. 10 shows that the Focus Estimator module is correctly able to predict the human's attention while they move around the environment.

5.2 Decision Tree Training

The Decision Tree that acts as the Movement Predictor is trained on a dataset generated in the simulated environment following the procedure described in Sect. 4.2. This dataset contains 362 training examples obtained from 10 random trials. The total number of samples per class is the following: STILL (46), WALK (97), PICK (30), TRANSPORT (161) and PLACE (28). The model fitted on this data is shown in Fig. 11.

To evaluate the fitness of this model, we have performed a 10-fold cross-validation: we split our dataset into the 10 groups from which it was generated, leave one aside for testing and train on the remaining, then repeat for each of the unique groups. Our average 10-fold cross-validation accuracy is 0.94.

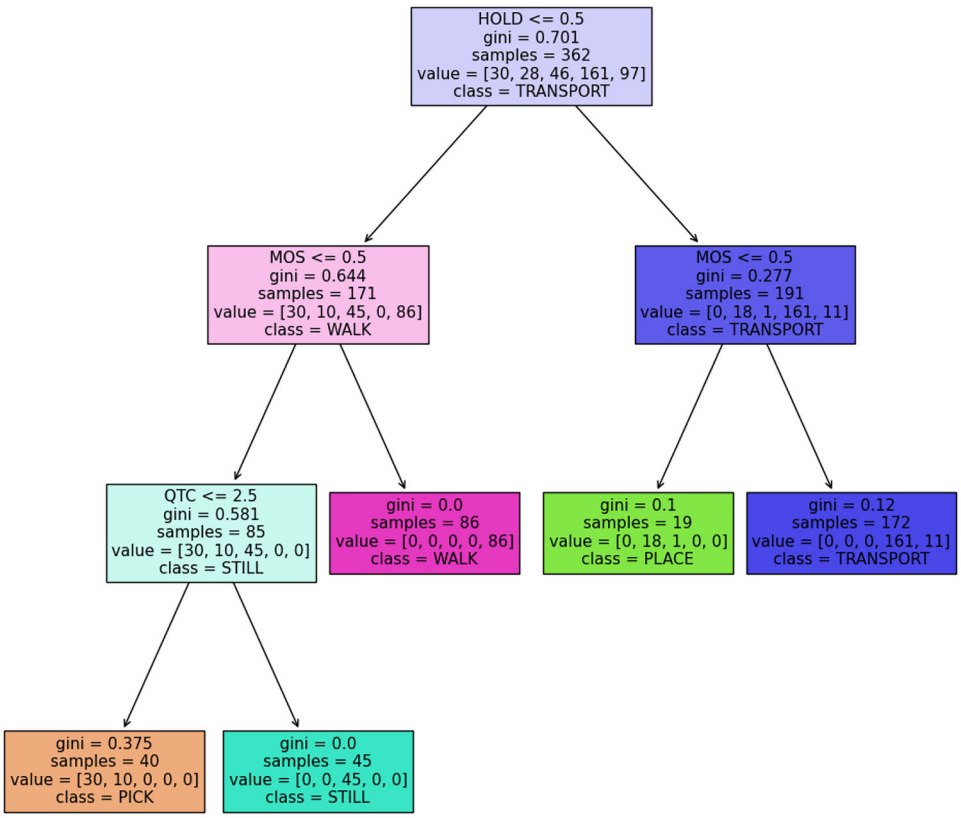
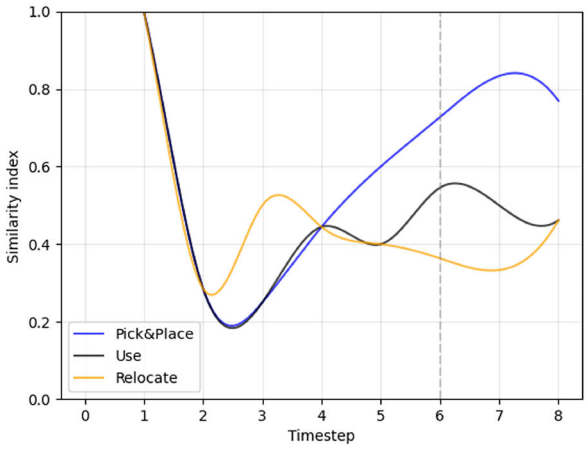
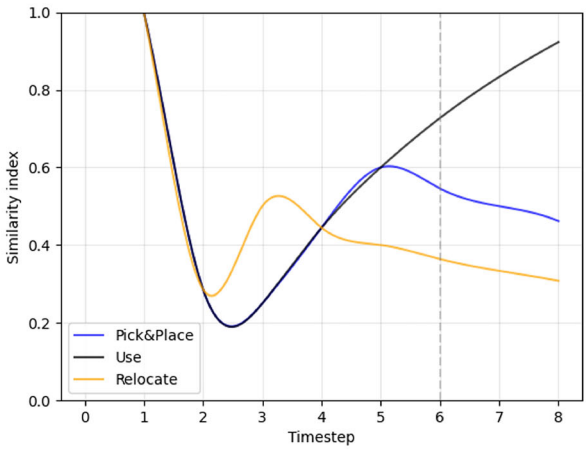


Fig. 11 The Decision Tree trained from the experimental data in the kitchen collaboration environment. The QSRs are classified as one of the following movements: Still, Walk, Transport, Pick or Place



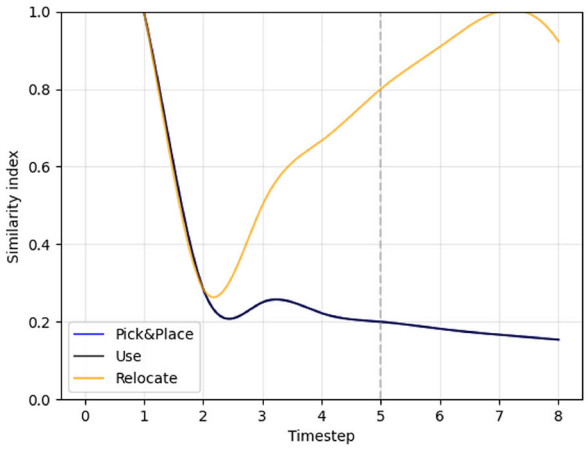
[STILL, WALK, PICK, TRANSPORT, PLACE, PICK, STILL, WALK, STILL]

(a)



[STILL, WALK, PICK, PLACE, PICK, PLACE, PICK, PLACE, STILL]

(b)



[STILL, WALK, STILL, WALK, STILL, WALK, STILL, WALK, STILL]

(c)

Fig. 12 Temporal response of the three FSMs which define the actions 'Pick and Place', 'Use' and 'Relocate' on different sequences of movements (reported below each graph). The vertical dashed line indicates the moment in which the ensemble has inferred the action

5.3 Markov Chain Finite-State Machines

To evaluate the fitness of the Action Predictor, we generate 3 sequences of 9 movements and we input them incrementally into the ensemble to analyze its temporal response to the observations. The results are reported in Fig. 12, where we have plotted the similarity score calculated through the Ratcliff-Obershelp algorithm at each timestep.

Since the sampling of each FSM uses the initial observed symbol as the starting state, the similarity score is maximum for the first iterations. Despite that, none of the models prevails on the others and no winner is declared yet. As soon as more symbols are fed into the system, the scores start to oscillate and differ, leaving one clear winner: this is the model that best describes the observed sequence of observations. The first two sequences produce a prediction on timestep 6, while the third one receives an inference on timestep 5. These predictions are in line with what we would expect given the symbols in the input sequences.

5.4 Explanation Generation

To test the Goal Reasoner embedded in the High-Level module, we have run 9 trials in which we have provided it with several sequences of observations. For the rest of this discussion, please refer to Fig. 8 for the structure of our Goal Library.

Table 4 summarizes the data we have collected. The latter shows, from left to right: the id of the trial, the observation (action) that was incrementally input in the system, the number of explanations generated by the reasoner, the time in microseconds required to produce the result, the confidence of the top-scoring explanation and finally the output of the component. If the latter is blank, then the reasoner could not formulate a prediction, otherwise it will report the name of the goal whose plan explains the observations. The system was reset between each trial.

Trial 1 presents to the High-Level the actions 'Pick and place' and 'Eat'. There are three possible explanations for these observations: one that describes the goal Breakfast with only observed and unobserved nodes and two that represent the plan for Lunch that accounts for several missing nodes (recall that an unobserved node is marked as missed if another node is observed on its right-hand side). Since our model assigns a higher score to the simplest model to describe the data, Breakfast is chosen as the prediction.

Trials 2 and 3 are straightforward: the goals Drink and Lunch are the only ones that contain respectively the actions 'Pick and Place' followed by either 'Sip' and 'Cook', so the Goal Reasoner can formulate a very confident prediction.

Trial 4 and 5 are more ambiguous: both of them begin with two 'Pick and place' actions which could describe each of the models with similar probability. Only when the system

Table 4 Experimental results on the goal reasoner

Trial	Actions	Explanations	Time (μs)	Confidence	Outcome
#1	Pick&Place	7	3.58	0.23	
	Eat	3	1.91	0.53	Breakfast
#2	Pick&Place	7	5.72	0.23	
	Sip	1	2.15	1	Drink
#3	Pick&Place	7	3.34	0.23	
	Cook	1	1.91	1	Lunch
#4	Pick&Place	7	3.58	0.23	
	Pick&Place	5	3.34	0.28	
	Eat	1	1.43	1	Lunch
#5	Pick&Place	7	4.29	0.23	
	Pick&Place	5	3.34	0.28	
	Pick&Place	1	1.19	1	Lunch
#6	Pick&Place	7	3.34	0.23	
	Pick&Place	5	2.62	0.28	
	Wash	5	2.62	0.3	
#7	Sip	1	3.1	1	Drink
#8	Eat	2	2.38	0.69	
#9	Wash	3	4.29	0.41	
	Cook	0	4.53	0	

See Sect. 5.4 for the full explanation

receives the third observation it can commit to a clear inference. Trial 6 follows a similar narrative, but the third input is still not able to disambuigate the goal: each of the possible plans share an uniform probability distribution and the reasoner fails to produce a prediction. No additional inputs would change the situation, since the rightmost node of the tree has been observed (remember that we assume that each observation happens after the preceding one). Of course, one could object that the goal could indeed by identified by the OOIs with which the human is interacting. This will be done in the full-scale experiment by the Verification component, which uses its ontology to filter out invalid explanations such as the goal Lunch if the first action 'Pick and Place' has been performed on the Biscuits.

In trial 7, we provide the system with a single observation that, alone, is able to discern the goal. We try doing the same with the observation 'Eat' in trial 8, but that action on its own could describe both Breakfast or Lunch. No further observation would be able to disambiguate this scenario, since the only next possible actions are 'Pick and Place' and 'Wash', which are common to both the candidates.

Finally, trial 9 shows an invalid sequence of actions: there is no goal plan in which the action 'Wash' is followed by any other action. For this reason, the system produced no valid explanations.

5.5 Intention Reading and Collaboration

Having verified the single components that constitute the Low-Level and High-Level modules of CASPER, we are now ready to analyze the overall performance of the cognitive architecture working together to read the human's intention and producing collaborative decision-making. The procedure we have followed is the following: we have run 5 trials for each of the 3 goals, collecting a total of 15 data samples. To test the system's robustness, at each iteration we have randomized the human's starting position and movement speed. For each trial, we have recorded: the number of observed and missed nodes in the winning explanation, the number of actions that the robot waits for the partner to complete before collaborating, the number of actions that the robots plans to execute, the accuracy of the prediction and the time, in seconds, needed to make an inference. The mean values of these variables are collected in Table 5.

The first thing to notice is that the robot was always able to correctly read the human's intention, despite some noise in the perceptual data collected by the system: the synergistic interaction of CASPER's components results in a robust intention reading performance. Table 5 also indicates that the time and observations required to infer the goal Lunch were higher than for the other two goals: this is in line with the higher complexity of its plan compared to the ones for Breakfast and Lunch.

Table 5 CASPER's performance on the kitchen experiment

Goal	Observed	Missed	Waiting	Planned	Accuracy	Time (s)
Breakfast	1.0	0.2	0.0	1.8	100%	42.21
Drink	1.0	0.4	0.0	1.6	100%	54.14
Lunch	1.8	0.2	1.8	2.0	100%	82.46

See Sect. 5.5 for the full explanation

The collaborative plan calculated from the cognitive architecture in each case was to wait for the human to finish eating their meal and then clean up the kitchen, which involves transporting the plate to the sink and washing it.

5.6 Verification

An additional experiment, this time in the form of an ablation study, was carried out to investigate the performance of the Verification module. In particular, we run the same experiment used to generate the data we have discussed in Sect. 5.5, this time disabling the formal verification of both the Low-Level and the High-Level: we call this the Non-Verified condition, as opposed as the Verified condition which represents the full cognitive architecture.

Figure 13 shows the comparison between the two configurations. The results that we collected prove that by disabling the Verification module we don't hurt the accuracy of the system, but instead we cause a slower prediction time. Despite the computational overhead introduced by the semantic reasoner, the Verified condition outperformed the Non-Verified one. This happens because the latter requires more observations to make sense of the environment, whilst the Verification module is able to discard noisy observations and illogical explanations before either of them are further processed, cutting down the overall inference time.

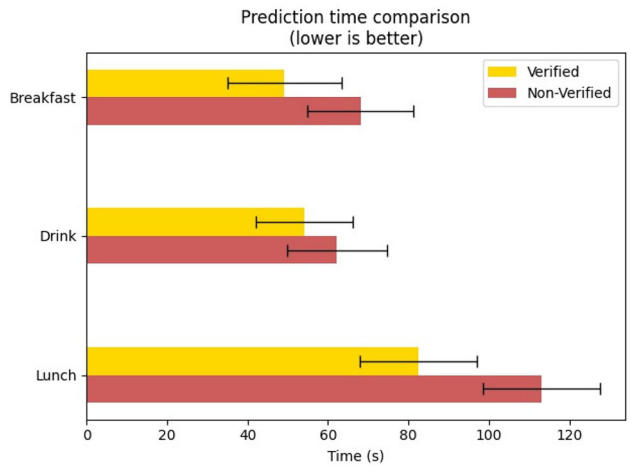


Fig. 13 Average prediction time of CASPER with and without the Verification module. Despite the computational overhead of the Pellet reasoner, the verification module ensures a faster inference from less observations

6 Conclusion and Future Work

6.1 Summary

In this paper, we have introduced CASPER: a symbolic cognitive architecture designed to perform intention reading and to calculate collaborative behaviors for human–robot teaming scenarios. Our system is able to accomplish the task through a set of parallel processes that communicate with each other and that can translate QSR descriptors into movements, then into actions and finally into goals and sub-goals using a bottom-up approach. Through the implementation of a simulated experimental case study based on a kitchen environment, we have empirically demonstrated the soundness of our methodologies.

The design of this system is driven by the requirement to embed in social robots the ability to autonomously integrate themselves in the structure of our daily routines, without the need for a human operator to explicitly provide instructions for the machine. Instead, by being able to understand the actions of other agents within the environment, a robot endowed with this cognitive architecture is able to seamlessly cooperate with them. In fact, despite our focus on human–robot interaction, this architecture would be equally applicable to robot-robot interactions.

Our main scientific contribution is the demonstration that QSRs can be used as an efficient means to achieve intention reading capabilities in artificial intelligence systems, a proof of concept that is lacking in the current state-of-the-art. Our technological contribution comes in the form of a cognitive architecture that incorporates novel algorithms for perception, reasoning and action selection which take inspiration from psychology and cognitive science.

6.2 Positioning in the Research Landscape

According to the taxonomy defined by Kotseruba and Tsotsos [2], CASPER is a symbolic architecture which implements the most common cognitive mechanisms, which we shall now summarize. Perception, the process that transforms raw input into the system's internal representation for carrying

out cognitive tasks, is performed by vision. Attention is modeled as a viewpoint/gaze selection mechanism: this means that the robot endowed with CASPER is able to select a target and track it through space. Action selection, which drives decision-making, is the result of a planning strategy which aims to maximize the relevance of the selected behavior. This architecture possesses all three types of memory: sensory (the QSR Library), working (handled by the Low- and High-Level) and long-term (in the form of the learned models and ontology). Learning comes in the form of declarative knowledge, which is a collection of facts about the world and various relationships defined between them. Reasoning is a cognitive ability which is present and central to each and every cognitive architecture, including this one. Finally, CASPER also implements metacognition, that is the ability to reason about one's own thoughts: this is done by the Verification module, which constantly monitors the other internal processes, identifying and correcting any erroneous decisions.

CASPER can be compared to some of the most renowned cognitive architectures designed for inferring human teammates' intentions. For instance, Scassellati [59] introduced a model of shared attention that enables the Cog robot to decode social cues such as gaze direction and pointing gestures. Both architectures share a fundamental assumption that intricate social skills can be deconstructed into simpler behavioral components, which can be more feasibly implemented on a robotic platform. Scassellati's work, in particular, delves into a division of joint attention into four developmental psychology-inspired stages: sustaining eye contact, tracking gaze, imperative pointing, and declarative pointing. Each stage builds on the preceding skills to develop increasingly complex behaviors. In contrast, CASPER performs intention reading by modularizing the cognitive skills required for perception, attention estimation, action recognition and prediction, goal composition and common-sense reasoning. The primary distinction between these two systems lies in the role of the human element. As postulated by Kanno et al. [60], the model from Scassellati performs "intended recognition", meaning that it assumes an active engagement of the partner, who is directly and explicitly involved in communicating with the robot through social cues. Conversely, CASPER adopts a form of observation often known as "keyhole recognition", where the human is not actively attempting to convey signals to the robot.

HAMMER (Hierarchical Attentive Multiple Models for Execution and Recognition) [61] employs a set of learned forward and inverse models to align world states with the motor actions needed to achieve or sustain them. This cognitive architecture serves both for executing actions and recognizing them when demonstrated by another agent. When applied across different hierarchical levels, this approach enables the robot to grasp the intentions of its partner. While this generative method is powerful, allowing for adaptable learning of actions and goals, its versatility comes at the expense of the generalizability that QSRs offer in terms of how each action can be performed and observed.

Finally, it is worth noting that CASPER adopts the cognitivist approach to cognition: it represents an hypothesis about those aspects of human cognition that are both relatively constant over time and independent of the task. In particular, it tries to achieve cognition by computations performed on internal symbolic knowledge representations. This stands in contrast to another category of cognitive architectures that embrace the emergent paradigm of cognitive science. In these models, the agent capitalizes on its embodiment to establish a close and dynamic connection between its sensorimotor system and the surrounding environment. One example of these architectures is EICA (Embodied Interactive Control Architecture) [62], which conceives intentions as interlinked processes, structured hierarchically and realized as dynamical systems. Both these approaches have their respective strengths and limitations, which could potentially be mitigated in the future through the development of hybrid cognitive architectures, combining the best attributes of both paradigms.

6.3 Limitations and Future Work

A limitation of our work is represented by the intrinsic nature of symbolic artificial intelligence: our methodologies suffer from the Knowledge Acquisition Bottleneck [63], which refers to the human intervention required to translate real-world conditions in symbolic inputs for the intelligent system. In our case, this comes in the form of the selection of OOIs, the plans for each goal and the ontology that envelops the properties of the environment, which have to be known a priori. Despite this, we argue that this disadvantage is compensated by the interpretability of each of the components that build up CASPER (the decisions of which can be explained at each step of computation, leading to potential increases in user trustworthiness and acceptability [64]) and the lack of computationally expensive and data-hungry processes.

Another limitation is given by the fact that we have tested CASPER in simulation and with fairly simple goals. The reason for this is that the work presented here is foundational to the true purpose of this cognitive architecture: that is, to offer support for intention reading and trust considerations in heterogeneous multi-agent teaming scenarios. Our planned future work involves initially testing the system's scalability: we aim to measure CASPER's accuracy and response time in more complex environments, incorporating a higher number of OOIs and an expanded collection of actions and movements. Subsequently, we plan to deploy CASPER in a scenario comprising multiple rooms, where distributed

groups of humans will interact with robots of varying make and capabilities to achieve a Team Goal. In this kind of setting, the individual robots will not be able to gather all the necessary information needed to predict the shared objective, rather they will have to rely on partial observations and communication with their peers. Moreover, performing empirical trials in simulation gives us the freedom to experiment with arbitrarily different environments, including varying numbers and type of agents.

It is important to remember that the purpose of CASPER is not to allow an agent to generalize across tasks, rather to equip the robot with a tool for performing appropriate collaborative actions in familiar environments. Integrating CASPER with another artificial cognitive architecture that models long-term memory learning, such as SOAR [9], might allow it to iteratively learn from unseen tasks. Another pathway might be the integration of Reinforcement Learning (RL). The latter has been used as a valid tool to create more generalizable and adaptive systems, bypassing the Knowledge Acquisition Bottleneck [63], but it often requires a large number of samples to learn optimal policies [65]. In addition, RL models can be challenging to interpret and explain, leading to a lack of transparency in decision-making that would conflict with one of the central design philosophies of CASPER.

Finally, it is worth mentioning one additional future expansion of CASPER: the inclusion of Artificial Trust (AT) [39, 66, 67]. By leveraging AT abilities, the robot will be able to assess the capabilities of other agents, whether humans or robots, to pursue the desired goal. Our hypothesis is that this cognitive skill will be valuable in enabling a group of heterogeneous robots to assign collaborative tasks among themselves effectively, thereby assisting the humans in their team.

Author Contributions S.V. designed the architecture, implemented the code, ran the experiments and wrote the manuscript. A.C. supervised the research and provided insights and guidance.

Funding This material is based upon work supported by the Air Force Office of Scientific Research, Air Force Materiel Command, USA. Funder award no. FA9550-19-1-7002. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising from this submission.

Data Availability Source code for this project is available in a public repository [68].

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethical Statement Not applicable: this work does not report on or involve the use of any animal or human data or tissue.

References

1. Vinanzi S, Goerick C, Cangelosi A (2019) Mindreading for robots: predicting intentions via dynamical clustering of human postures. In: 2019 Joint IEEE 9th international conference on development and learning and epigenetic robotics (ICDL-EpiRob), pp 272–277
2. Kotseruba I, Tsotsos JK (2020) 40 years of cognitive architectures: core cognitive abilities and practical applications. Artif Intell Rev 53(1):17–94
3. Cangelosi A, Asada M (2022) Cognitive robotics. MIT Press, Cambridge
4. Friedenberg J, Silverman G, Spivey MJ (2021) Cognitive science: an introduction to the study of mind. Sage Publications, Thousand Oaks
5. Tsotsos JK (2017) Attention and cognition: principles to guide modeling. In: Computational and cognitive neuroscience of vision. Springer, Singapore, pp 277–295. https://doi.org/10.1007/978-981-10-0213-7_12
6. Faghihi U, Poirier P, Larue O (2011) Emotional cognitive architectures. In: International conference on affective computing and intelligent interaction. Springer, pp 487–496
7. Epstein SL (2004) Metaknowledge for autonomous systems. In: Proceedings of AAAI spring symposium on knowledge representation and ontology for autonomous systems. AAAI, pp 61–68
8. Anderson JR (2013) The architecture of cognition. Psychology Press, New York
9. Laird JE, Newell A, Rosenbloom PS (1987) Soar: an architecture for general intelligence. Artif Intell 33(1):1–64
10. JSnaider J, McCall R, Franklin S (2011) The LIDA framework as a general tool for AGI. In: International conference on artificial general intelligence. Springer, pp 133–142
11. Wang P (1995) Non-axiomatic reasoning system: exploring the essence of intelligence. Indiana University, Bloomington
12. Chella A, Lanza F, Seidita V (2019) A cognitive architecture for human–robot teaming interaction. In: 6th international workshop on artificial intelligence and cognition, AIC 2018, vol 2418. CEUR-WS, pp 82–89
13. Han JH, Kim JH (2010) Human–robot interaction by reading human intention based on mirror-neuron system. In: 2010 IEEE international conference on robotics and biomimetics. IEEE, pp 561–566
14. Melis AP (2013) The evolutionary roots of human collaboration: coordination and sharing of resources. Ann NY Acad Sci 1299(1):68–76
15. El Makrini I, Merckaert K, Lefeber D, Vanderborght B (2017) Design of a collaborative architecture for human–robot assembly tasks. In: IROS, pp 1624–1629
16. Wang L, Gao R, Váncza J, Krüger J, Wang XV, Makris S, Chryssolouris G (2019) Symbiotic human–robot collaborative assembly. CIRP Ann 68(2):701–726

17. Tsarouchi P, Matthaiakis A-S, Makris S, Chryssolouris G (2017) On a human–robot collaboration in an assembly cell. Int J Comput Integr Manuf 30(6):580–589
18. Hoffman G, Breazeal C (2004) Collaboration in human–robot teams. In: AIAA 1st intelligent systems technical conference, p 6434
19. Bluethmann W, Ambrose R, Diftler M, Askew S, Huber E, Goza M, Rehnmark F, Lovchik C, Magruder D (2003) Robonaut: a robot designed to work with humans in space. Auton Robot 14(2–3):179–197
20. Lallée S, Yoshida E, Mallet A, Nori F, Natale L, Metta G, Warneken F, Dominey PF (2010) Human–robot cooperation based on interaction learning. In: From motor learning to interaction learning in robots. Springer, pp 491–536
21. Pineau J, Montemerlo M, Pollack M, Roy N, Thrun S (2003) Towards robotic assistants in nursing homes: challenges and results. Robot Auton Syst 42(3–4):271–281
22. Semeraro F, Griffiths A, Cangelosi A (2023) Human-robot collaboration and machine learning: a systematic review of recent research. Robot Comput Integr Manuf 79:102432. https://doi.org/10.11016/j.rcim.2022.102432
23. Woodward AL, Sommerville JA, Gerson S, Henderson AME, Buresh J (2009) The emergence of intention attribution in infancy. Psychol Learn Motiv 51:187–222
24. Malle BF, Moses LJ, Baldwin DA (2001) Intentions and intentionality: foundations of social cognition. MIT Press, Cambridge
25. Baldwin DA, Baird JA (2001) Discerning intentions in dynamic human action. Trends Cogn Sci 5(4):171–178
26. Tomasello M, Carpenter M, Call J, Behne T, Moll H (2005) Understanding and sharing intentions: the origins of cultural cognition. Behav Brain Sci 28(5):675–691
27. Dominey PF, Warneken F (2011) The basis of shared intentions in human and robot cognition. New Ideas Psychol 29(3):260–274
28. Duarte NF, Raković M, Tasevski J, Coco MI, Billard A, Santos-Victor J (2018) Action anticipation: reading the intentions of humans and robots. IEEE Robot Autom Lett 3(4):4132–4139
29. Zenn Bien Z, Park K-H, Jung J-W, Do J-H (2005) Intention reading is essential in human-friendly interfaces for the elderly and the handicapped. IEEE Trans Ind Electron 52(6):1500–1505
30. Tahboub KA (2006) Intelligent human–machine interaction based on dynamic Bayesian networks probabilistic intention recognition. J Intell Robot Syst 45(1):31–52
31. Buonamente M, Dindo H, Johnsson M (2013) Recognizing actions with the associative self-organizing map. In: 2013 XXIV international symposium on information, communication and automation technologies (ICAT). IEEE, pp 1–5
32. Jansen B, Belpaeme T (2006) A computational model of intention reading in imitation. Robot Auton Syst 54(5):394–402
33. Oztop E, Kawato M, Arbib M (2006) Mirror neurons and imitation: a computationally guided review. Neural Netw 19(3):254–271
34. Granada RL, Pereira RF, Monteiro J, Ruiz DD, Barros RC, Meneguzzi FR(1995) Hybrid activity and plan recognition for video streams. In: Greer, JE and Koehn, GM (eds) The 2017 AAAI workshop on plan, activity, and intent recognition, pp 54–59
35. Fod A, Matarić MJ, Jenkins OC (2002) Automated derivation of primitives for movement classification. Auton Robot 12:39–54
36. Krüger V, Kragic D, Ude A, Geib C (2007) The meaning of action: a review on action recognition and mapping. Adv Robot 21(13):1473–1501
37. Geib CW, Goldman RP (2009) A probabilistic plan recognition algorithm based on plan tree grammars. Artif Intell 173(11):1101–1132
38. Mirsky R, Gal Y, Shieber SM (2017) Cradle: an online plan recognition algorithm for exploratory domains. ACM Trans Intell Syst Technol (TIST) 8(3):1–22
39. Vinanzi S, Cangelosi A, Goerick C (2021) The collaborative mind: intention reading and trust in human–robot interaction. iScience 24(2):102130. https://doi.org/10.1016/j.isci.2021.102130
40. Vinanzi S, Cangelosi A, Goerick C (2020) The role of social cues for goal disambiguation in human–robot cooperation. In: 2020 29th IEEE international conference on robot and human interactive communication (RO-MAN), pp 971–977
41. Moratz R (2017) Qualitative spatial reasoning. Springer, Cham, pp 1700–1707
42. Duckworth P, Hogg DC, Cohn AG (2019) Unsupervised human activity analysis for intelligent mobile robots. Artif Intell 270:67–92
43. Ardhendu Behera, Anthony G Cohn, and David C Hogg (2012) Workflow activity monitoring using dynamics of pair-wise qualitative spatial relations. In: International Conference on Multimedia Modeling, pages 196–209. Springer
44. Dubba K, de Oliveira M, Lim GH, Kasaei H, Lopes LS, Tomé A, Cohn A (2014) Grounding language in perception for scene conceptualization in autonomous robots. In: 2014 AAAI spring symposium series
45. Young J, Hawes N (2015) Learning by observation using qualitative spatial relations. In: International conference on autonomous agents & multiagent systems, 2015. Association for Computing Machinery, pp 745–751
46. Krishnaswamy N, Friedman S, Pustejovsky J (2019) Combining deep learning and qualitative spatial reasoning to learn complex structures from sparse examples with noise. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 2911–2918
47. Gatsoulis Y, Alomari M, Burbridge C, Dondrup C, Duckworth P, Lightbody P, Hanheide M, Hawes N, Hogg DC, Cohn AG, et al (2016) Qsrlib: a software library for online acquisition of qualitative spatial relations from video. In: Workshop on qualitative reasoning (QR16), at IJCAI-16
48. Clementini E, Di Felice P, Hernández D (1997) Qualitative representation of positional information. Artif Intell 95(2):317–356
49. Delafontaine M, Cohn AG, Van de Weghe N (2011) Implementing a qualitative calculus to analyse moving point objects. Expert Syst Appl 38(5):5187–5196
50. Van de Weghe N (2004) Representing and reasoning about moving objects: a qualitative approach. PhD thesis, Ghent University
51. Chong E, Ruiz N, Wang Y, Zhang Y, Rozga A, Rehg JM (2018) Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency. In: Proceedings of the European conference on computer vision (ECCV), pp 383–398
52. Kotsiantis SB (2013) Decision trees: a recent overview. Artif Intell Rev 39:261–283
53. Vidal E, Thollard F, De La Higuera C, Casacuberta F, Carrasco RC (2005) Probabilistic finite-state machines-part I. IEEE Trans Pattern Anal Mach Intell 27(7):1013–1025
54. Manzi A, Dario P, Cavallo F (2017) A human activity recognition system based on dynamic clustering of skeleton data. Sensors 17(5):1100
55. Ratcliff JW, Metzener DE (1988) Pattern-matching-the gestalt approach. Dr Dobbs J 13(7):46
56. Parsia B, Sirin E (2004) Pellet: an owl dl reasoner. In: Third international semantic web conference-poster, vol 18. Citeseer, p 13
57. Michel Olivier (2004) Cyberbotics ltd. webots™: professional mobile robot simulation. Int J Adv Robot Syst 1(1):5
58. Horrocks I, Patel-Schneider PF, Boley H, Tabet S, Grosof B, Dean M et al (2004) SWRL: A semantic web rule language combining OWL and RuleML. W3C Memb Submiss 21(79):1–31
59. Scassellati B (1998) Imitation and mechanisms of joint attention: a developmental structure for building social skills on a humanoid robot. In: International workshop on computation for metaphors, analogy, and agents. Springer, pp 176–195

60. Kanno T, Nakata K, Furuta K (2003) A method for team intention inference. Int J Hum Comput Stud 58(4):393–413
61. Demiris Y, Khadhouri B (2006) Hierarchical attentive multiple models for execution and recognition of actions. Robot Auton Syst 54(5):361–369
62. Mohammad YFO, Nishida T (2007) Intention through interaction: toward mutual intention in real world interactions. In: New trends in applied artificial intelligence: 20th international conference on industrial, engineering and other applications of applied intelligent systems, IEA/AIE 2007, Kyoto, Japan, June 26–29, 2007. Proceedings, vol 20. Springer, pp 115–125
63. Cullen JABRYMAN, Bryman A (1988) The knowledge acquisition bottleneck: time for reassessment? Expert Syst 5(3):216–225
64. Shin D (2021) The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable AI. Int J Hum Comput Stud 146:102551
65. Qi S, Zhu S-C (2018) Intent-aware multi-agent reinforcement learning. In 2018 IEEE international conference on robotics and automation (ICRA), pp 7533–7540
66. Vinanzi S, Patacchiola M, Chella A, Cangelosi A (2019) Would a robot trust you? Developmental robotics model of trust and theory of mind. Philos Trans R Soc B Biol Sci 374(1771):20180032. https://doi.org/10.1098/rstb.2018.0032
67. Jorge CC, Tielman ML, Jonker CM (2022) Artificial trust as a tool in human-AI teams. In: Proceedings of the 2022 ACM/IEEE international conference on human–robot interaction, pp 1155–1157
68. Vinanzi S (2023) Casper source code public repository. https://github.com/samvinanzi/CASPER

Samuele Vinanzi is a Senior Lecturer in Robotics and Artificial Intelligence at Sheffield Hallam University, United Kingdom. He earned his BSc and MSc in Computer Engineering from the University of Palermo, and his Ph.D. from the University of Manchester. Specializing in Cognitive Robotics, his research focuses on developing intelligent robotic systems capable of perceiving, reasoning, and interacting with the environment in a human-like manner. His work, sponsored by the Honda Research Institute Europe and the U.S. Air Force Office of Scientific Research, centers on computational models for intention reading, Artificial Trust, and collaboration in humanoid robots.

Angelo Cangelosi is Professor of Machine Learning and Robotics at the University of Manchester (UK) and co-director and founder of the Manchester Centre for Robotics and AI. He was selected for the award of the European Research Council (ERC) Advanced grant (UKRI funded). His research interests are in cognitive and developmental robotics, neural networks, language grounding, human robot-interaction and trust, and robot companions for health and social care. Overall, he has secured over £40m of research grants as coordinator/PI, including the ERC Advanced eTALK, the UKRI TAS Trust Node and CRADLE Prosperity, the US AFRL project THRIVE++, and numerous Horizon and MSCAs grants. Cangelosi has produced more than 300 scientific publications. He is Editor-in-Chief of the journals Interaction Studies and IET Cognitive Computation and Systems, and in 2015 was Editor-in-Chief of IEEE Transactions on Autonomous Development. He has chaired numerous international conferences, including ICANN2022 Bristol, and ICDL2021 Beijing. His book "Developmental Robotics: From Babies to Robots" (MIT Press) was published in January 2015, and translated in Chinese and Japanese. His latest book "Cognitive Robotics" (MIT Press), coedited with Minoru Asada, was recently published in 2022.